

ADLIB SDK - Swift 연동 가이드 v.1.0

: 스위프트 개발환경에서 ADLIB SDK를 연동하기 위한 가이드 문서

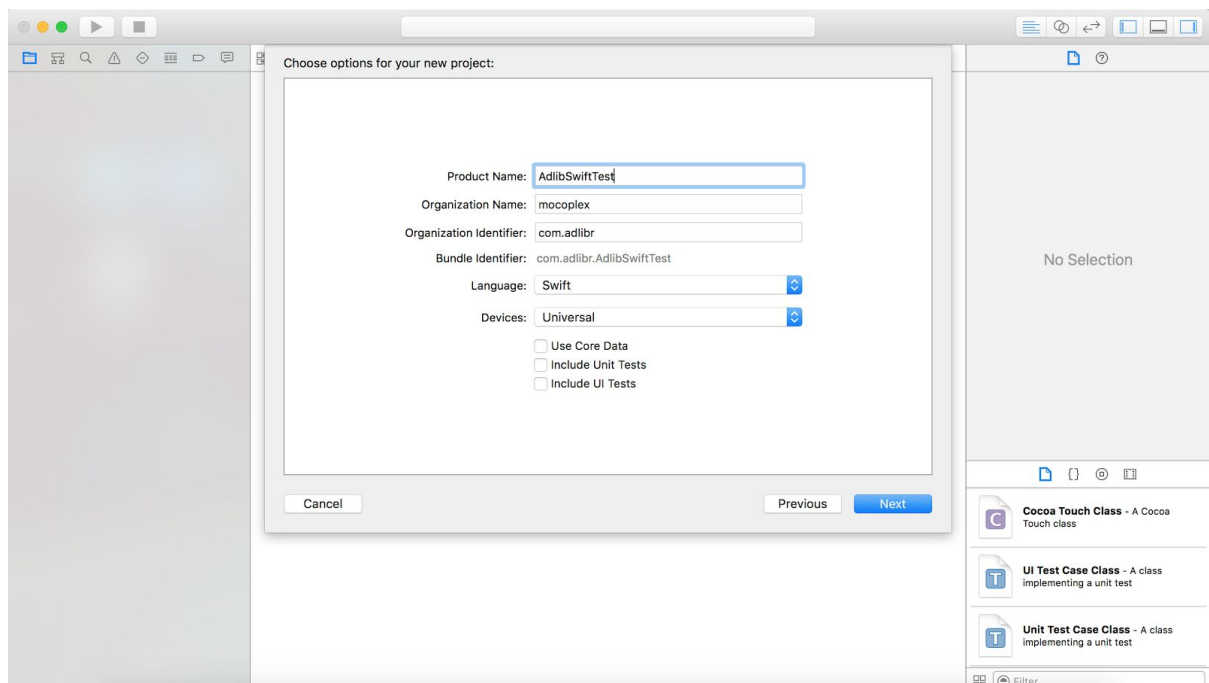
문의 : gskang@mocoplex.com

0. 개발 환경

- XCode 7.3 (7D175)
- Swift 2.2

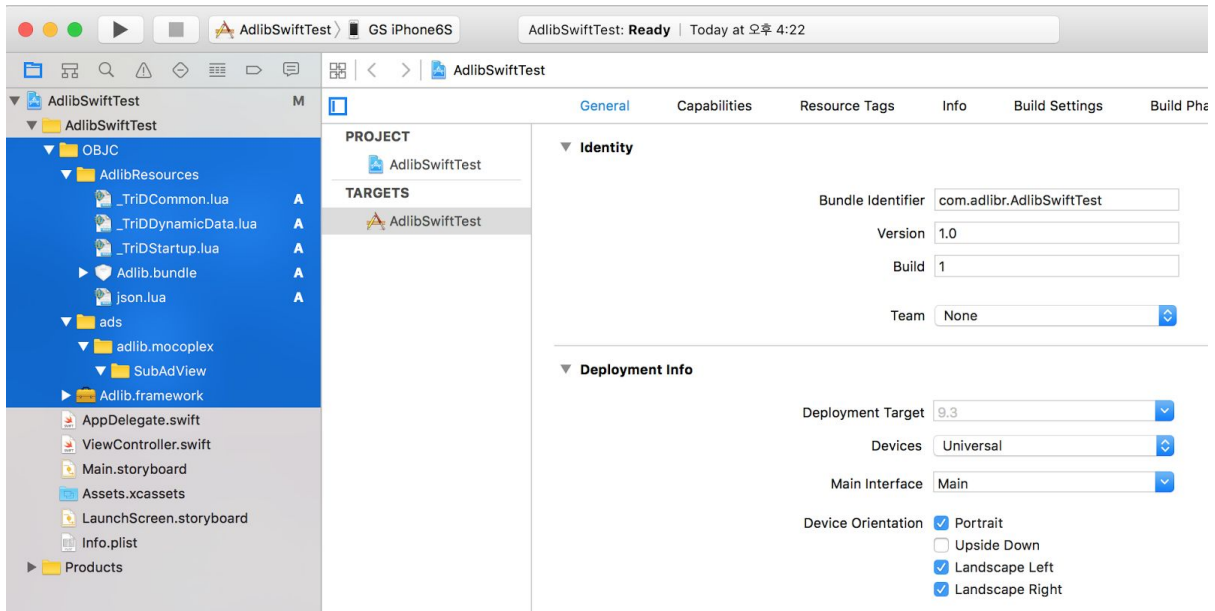
1. 프로젝트 설정

1.1 Swift 프로젝트 생성



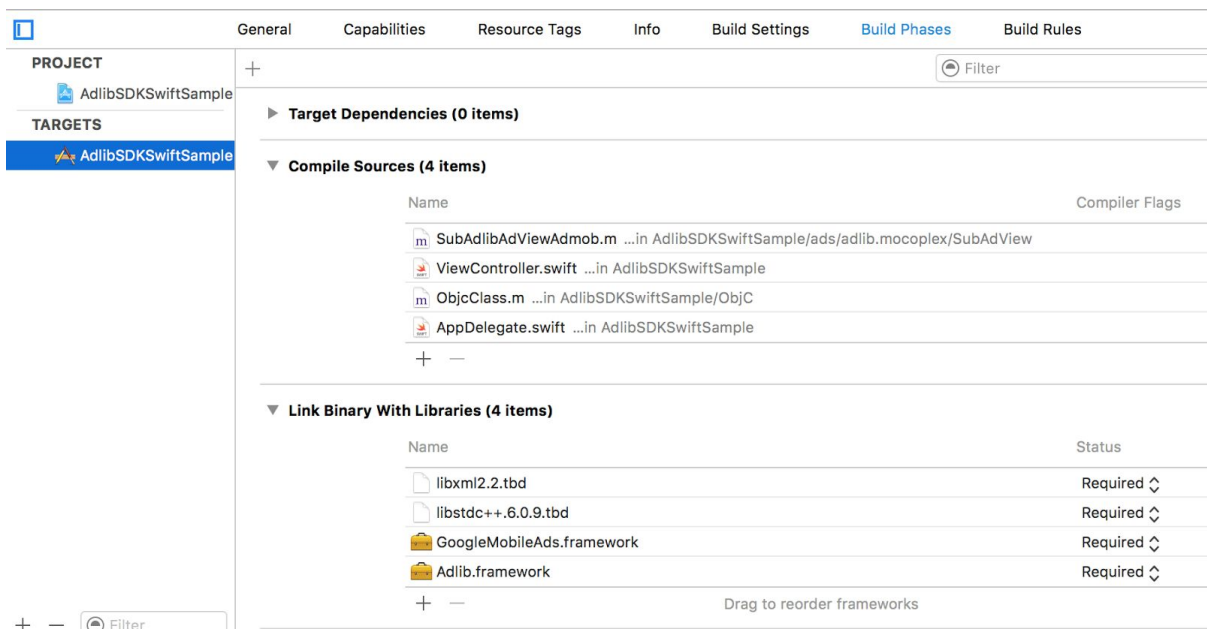
1.2 애드립 SDK 및 리소스 파일 추가

애드립 광고 노출을 위해 필요한 Adlib.framework 파일과 AdlibResource폴더의 파일들을 프로젝트에 추가합니다. 사용하실 플랫폼의 SubAdView 폴더 내부의 클래스 파일들과 해당 플랫폼의 SDK 또한 추가합니다. 제공드린 샘플 프로젝트에는 애드립과 애드몹 추가에 필요한 리소스 파일들이 추가되어 있습니다.



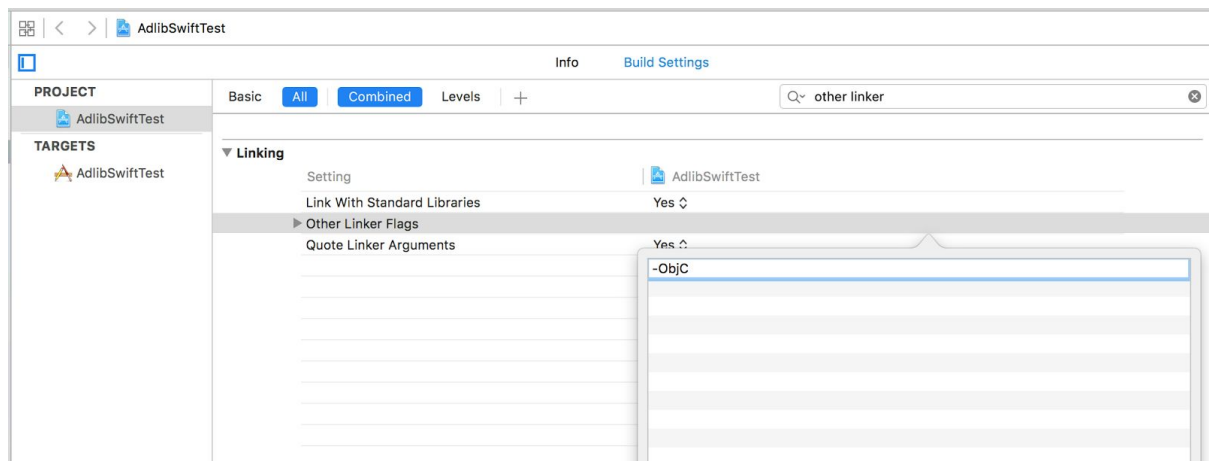
1.3 프로젝트 빌드 Phases 라이브러리 추가

하단 이미지처럼 libxml2.2.tbd / libstdc++.6.0.9.tbd 을 추가하시고, Adlib.framework 파일과
사용하실 광고 플랫폼의 라이브러리 추가 내용을 확인합니다. (샘플 프로젝트 기준 애드몹
framework파일이 추가되어 있습니다.)



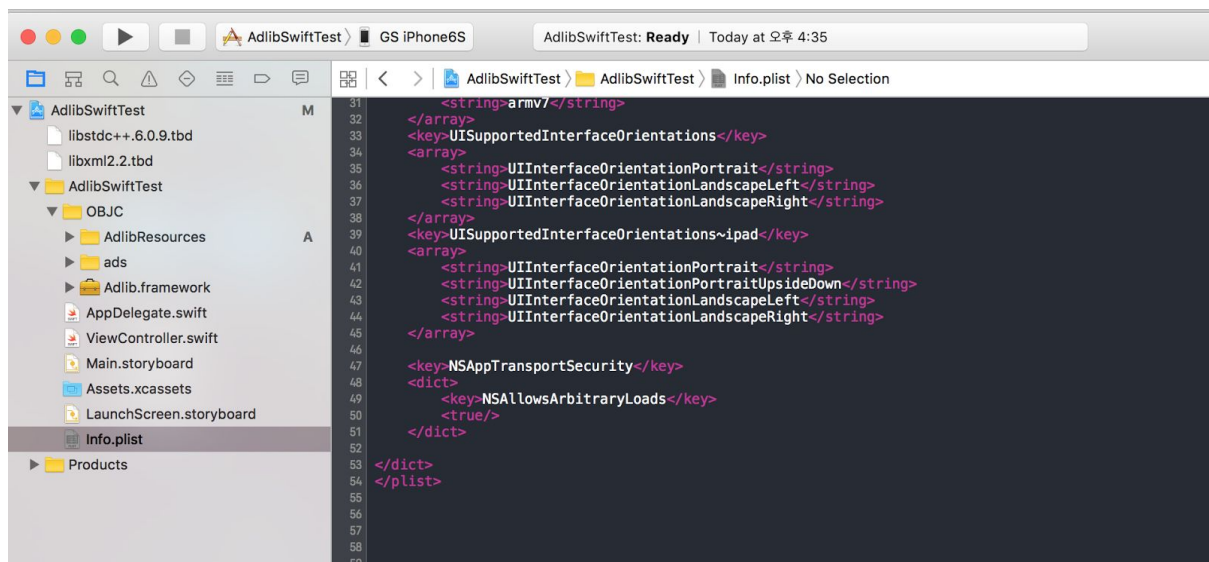
1.4 프로젝트 Build Settings 링커 플러그 추가

하단의 이미지 처럼 Other Linker Flags 항목에 -ObjC 플러그를 추가합니다.



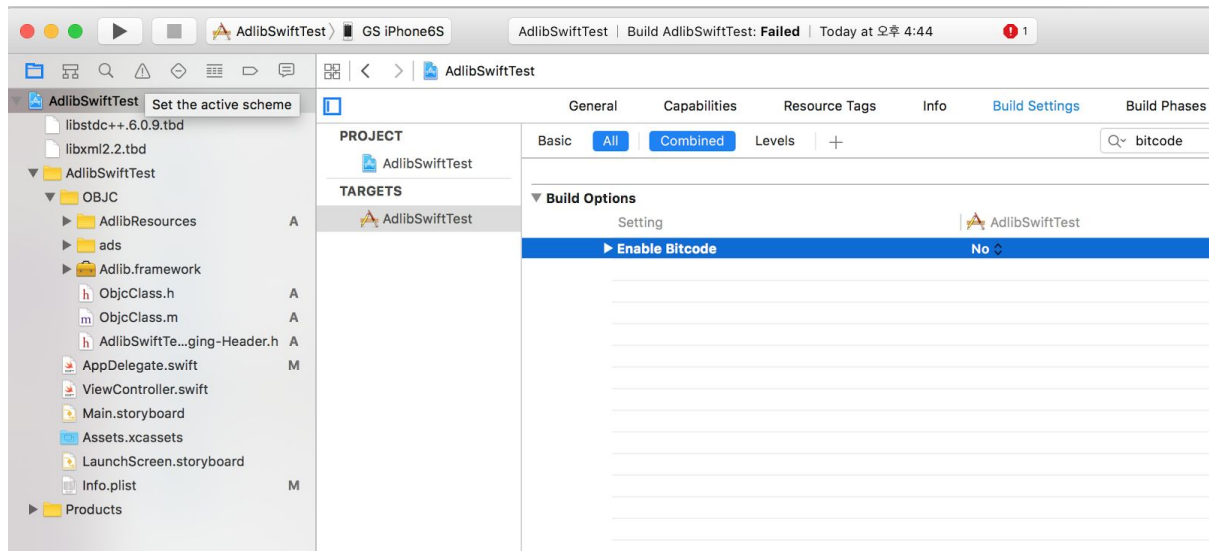
1.5 프로젝트 ATS 설정

프로젝트 Info.plist 파일 소스 보기 항목으로 ATS 관련 설정 값을 추가합니다.



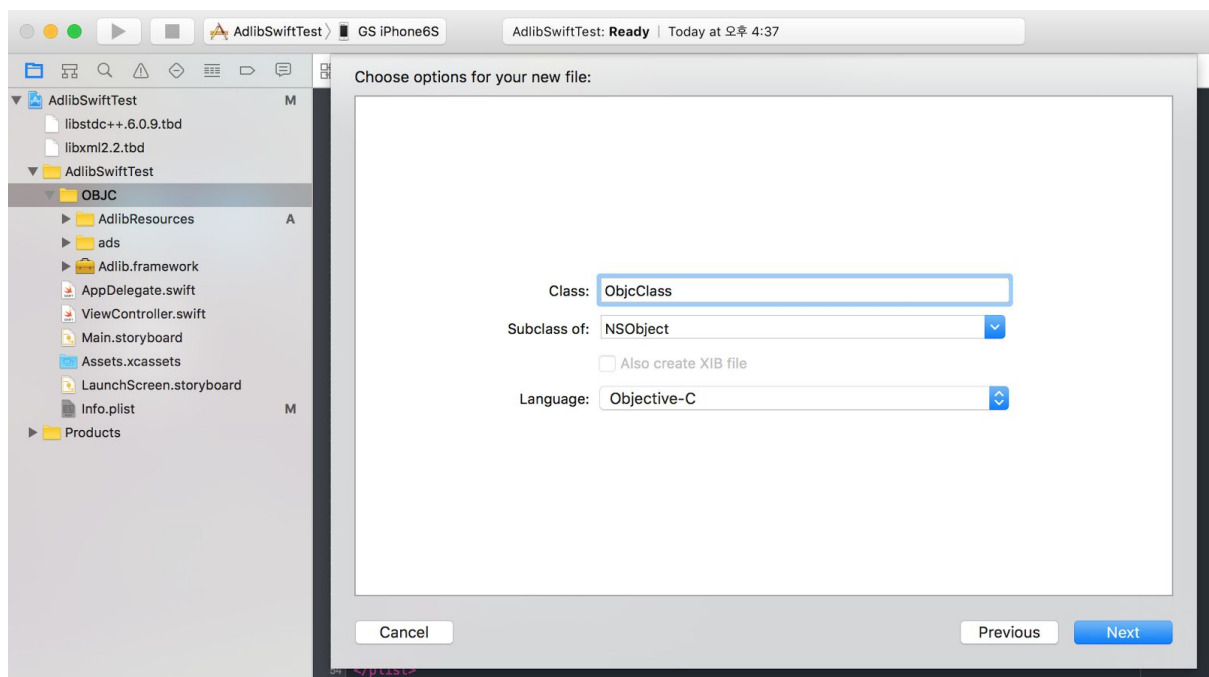
1.6 Bitcode 설정

bitcode 설정을 NO로 변경합니다. (bitcode 컴파일이 필요한 상황의 경우 별도로 bitcode 적용이 가능한 SDK 요청이 필요합니다.)

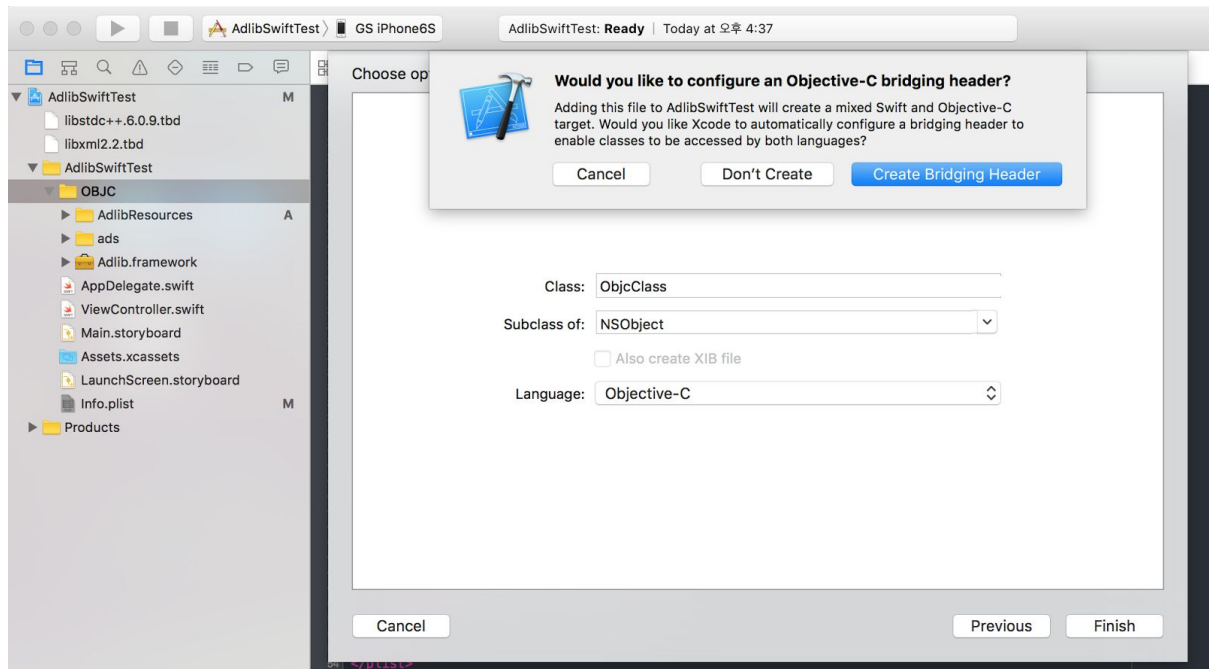


1.7 ObjC -Swift Bridge Header 추가

스위프트 프로젝트에서 ObjectiveC 코드를 사용하기 위해 필요한 브릿지 헤더를 추가하는 작업을 수행합니다. 하단의 이미지 처럼 프로젝트에 클래스를 추가하면서 Language 항목을 Objective-C 로 변경하여 클래스를 추가합니다.

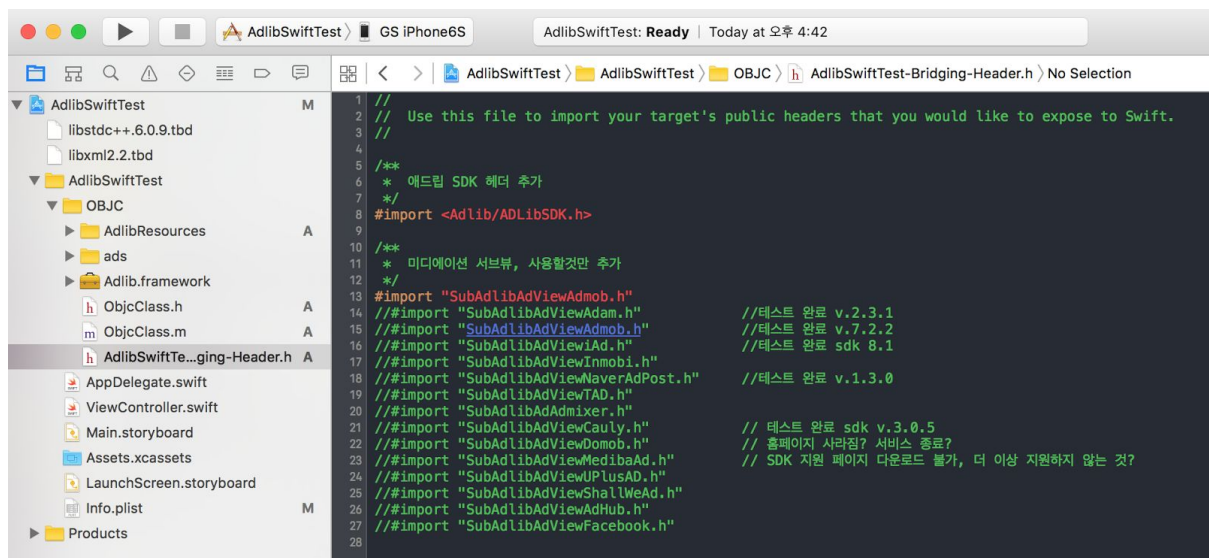


이미지 처럼 팝업이 누출되면, Create Bridging Header 버튼을 선택하면, 자동으로 브릿지 파일을 추가됩니다.



1. 8 브릿지 파일 처리

브릿지 파일에 하단 이미지 (샘플 프로젝트 코드 참고)처럼 내용을 추가합니다.



AdlibSwiftTest-Bridging-Header

```
//
// Use this file to import your target's public headers that you would like to expose
// to Swift.
//

/**
 * 애드립 SDK 헤더 추가
 */
#import <Adlib/ADLibSDK.h>

/**
 * 미디어이션 서브뷰, 사용할것만 추가
 */
// #import "SubAdlibAdViewAdmob.h"
// #import "SubAdlibAdViewAdam.h" //테스트 완료 v.2.3.1
// #import "SubAdlibAdViewAdmob.h" //테스트 완료 v.7.2.2
// #import "SubAdlibAdViewiAd.h" //테스트 완료 sdk 8.1
// #import "SubAdlibAdViewInmobi.h"
// #import "SubAdlibAdViewNaverAdPost.h" //테스트 완료 v.1.3.0
// #import "SubAdlibAdViewTAD.h"
// #import "SubAdlibAdAdmixer.h"
// #import "SubAdlibAdViewCauly.h" // 테스트 완료 sdk v.3.0.5
// #import "SubAdlibAdViewDomob.h"
// #import "SubAdlibAdViewMedibaAd.h"
// #import "SubAdlibAdViewUPlusAD.h"
// #import "SubAdlibAdViewShallWeAd.h"
// #import "SubAdlibAdViewAdHub.h"
// #import "SubAdlibAdViewFacebook.h"
```

2. 애드립 초기화

하단 코드 및 샘플프로젝트의 AppDelegate.swift 파일의 구현을 참고하여, 발급 받으신 애드립 키 값으로 설정 초기화를 수행합니다.

AppDelegate.swift

```
@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate, AdlibSessionDelegate {

    var window: UIWindow?

    func application(application: UIApplication, didFinishLaunchingWithOptions
launchOptions: [NSObject: AnyObject]?) -> Bool {

        // 애드립 키로 초기화
        // 실제 사용하실 키로 교체하여 진행하시길 바랍니다.
        // 테스트 키세팅 띠배너 미디어이션 :(애드립, 애드몹, 아담)
        // 테스트 키세팅 전면배너 미디어이션 :(애드립, 애드몹, 아담, 애드립하우스)

        let adlibKey: String = "54caefb80cf28de2af4e7001"
```

```

let sharedManager = AdlibManager.sharedSingletonClass()

//애드립 델리게이트 설정
sharedManager.sessionDelegate = self

//테스트, 사용 모드 설정
let isTestMode = true

if isTestMode {
    sharedManager.testModeLinkWithAdlibKey(adlibKey)
} else {
    sharedManager.linkWithAdlibKey(adlibKey)
}

return true
}

//애드립 설정 정보 델리게이트 구현
func adlibManager(manager: AdlibManager!, didLinkedSessionWithUserInfo userInfo:
[NSObject : AnyObject]!) {
    print("adlib linked")
}

func adlibManager(manager: AdlibManager!, didFailedSessionLinkWithError error:
NSError!) {
    print("adlib link failed")
}
}

```

3. 띠배너 요청

자세한 구현은 샘플 프로젝트의 ViewController.swift 파일을 참고합니다.

뷰컨트롤러 class에 AdlibManagerDelegate 구현을 추가합니다.

뷰의 노출 / 삭제 시점에 AdlibManager 클래스를 통해 광고를 적재할 뷰위에 attach / detach 메소드를 호출합니다.

```

class ViewController: UIViewController, AdlibManagerDelegate {

    @IBOutlet weak var adView: UIView?

    override func viewDidLoad() {
        super.viewDidLoad()
    }

    override func viewDidAppear(animated: Bool) {

        let sharedManager = AdlibManager.sharedSingletonClass()

        // adView 띠배너 광고를 적재할 부모 뷰 (컨테이너 뷰)
        // 샘플프로젝트 상의 Storyboard에서 확인
        sharedManager.attach(self, withView: adView, withDelegate: self)
    }
}

```

```

override func viewDidDisappear(animated: Bool) {

    let sharedManager = AdlibManager.sharedSingletonClass()
    sharedManager.detach(self)
}

//애드립 띠배너 Delegate 처리
func didReceiveAdlibAd(from: String!) {
    print("received adlib Ad")
}

func didFailToReceiveAdlibAd(from: String!) {
    print("didFailed adlib Ad")
}

//애드립 전면배너 Delegate 처리
func didReceiveAdlibInterstitialAd(from: String!) {
    print("received adlib interstitial Ad")
}

func didFailToReceiveAllInterstitialAd() {
    print("failed to load adlib interstitial Ad")
}
}

```

애드립 델리게이트는 띠 / 전면 배너의 수신 성공, 실패에 대한 로그를 제공합니다.
노출에 대한 처리는 내부적으로 수행되며 델리게이트 상에서의 특별한 처리는 필요하지 않습니다.

3. 전면 배너 요청

```

class ViewController: UIViewController, AdlibManagerDelegate {

    @IBOutlet weak var adView: UIView?

    override func viewDidLoad() {
        super.viewDidLoad()
    }

    //전면광고 요청 버튼처리
    @IBAction func actionButton() {

        let sharedManager = AdlibManager.sharedSingletonClass()
        sharedManager.delegate = self
        sharedManager.loadInterstitialAd(self, withDelegate: self)
    }

    //애드립 전면배너 Delegate 처리
    func didReceiveAdlibInterstitialAd(from: String!) {
        print("received adlib interstitial Ad")
    }

    func didFailToReceiveAllInterstitialAd() {
        print("failed to load adlib interstitial Ad")
    }
}

```



```
}  
}
```

4. 마무리

스위프트 연동을 제외한 애드립 연동에 대한 가이드는 하단 링크페이지를 참고합니다.

http://developer.adlibr.com/ssp_ios_guide.html

기타 연동에 필요한 문의 사항이 존재 시 하단 페이지로 문의 부탁 드립니다.

<https://groups.google.com/forum/?hl=ko&fromgroups#!forum/adlib-network>