适用范围 ： 17K 作品内容相关信息的输出

接入流程 ： 申请颁发私钥和 ID 配合 Bundle ldentifier 使用

iOS SDK 集成

一、集成

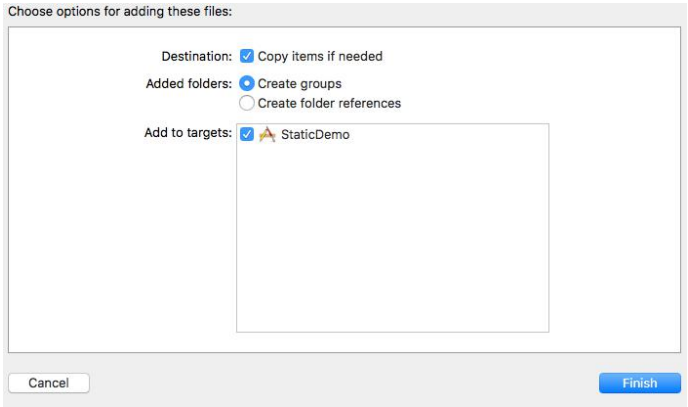
SDK 为 Swift 开发，需要 swift 项目或者 oc-swift 混编项目，纯 oc 项目需要改成支持 swift（混编）

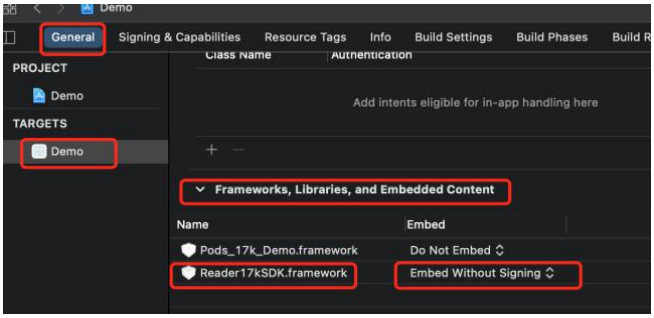
方式一 推荐 pod 集成

```
pod 'Reader17k', :path => './LocalPod/Reader17k'
```

方式二 xcframework 集成

将 Reader17kSDK.xcframework 引入项目





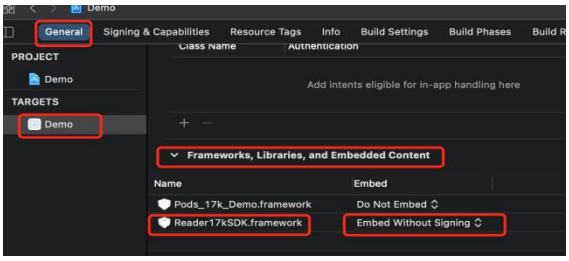
app 集成后增量:

iOS9+ 增量 5.1M （armv7 + arm64）

二、问题

1. dyld: Library not loaded: @rpath/XXXX Reason:XXX 问题

需要把库加入到工程中，需要 Embed

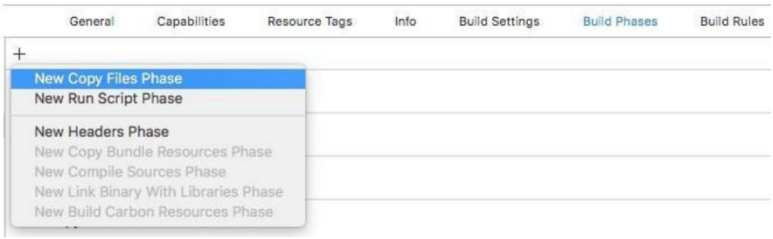


老项目参考下面方法:

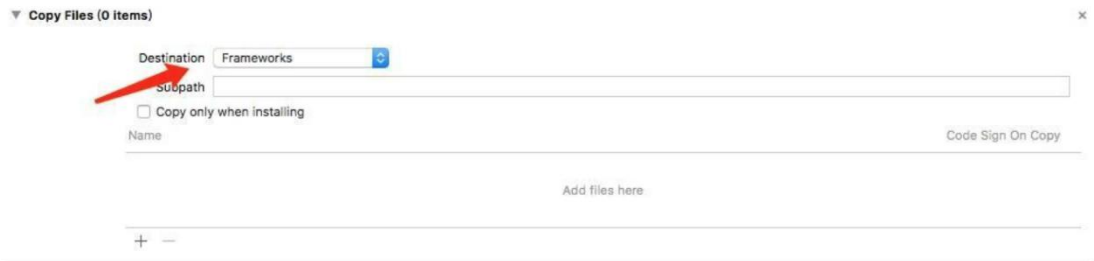
如果有 Embed Frameworks，那么就直接加入即可

如果没有的话

1>选择新增 New Copy Files Phase



2>将 Destination 设置为 Frameworks



3>加入对应的动态库

2. 网络链接失败,有部分链接使用的是 http 请求需要在 info 中配置

1. 在 Info.plist 中添加 NSAppTransportSecurity 类型 Dictionary。
2. 在 NSAppTransportSecurity 下添加 NSAllowsArbitraryLoads 类型 Boolean,值设为 YES

三、使用说明（详情参考 demo）

swift

```swift
import Reader17kSDK
```

oc

```objc
#import <Reader17kSDK/Reader17kSDK.h>
```

配置 SDK

```swift
/// 配置SDK
///
/// - Parameters:
///    - appid:  渠道 申请获得
///    - secret: 渠道 申请获得
///    - options: 其它信息{
///                          "enableService" = " (0 或 1) " //是否显示联系客服入口 default 1
///                          "servcie" = "urlStr" // 客服链接地址 默认使用SDK客服地址
///                      }
///    }
@objc public func config(appid: String, secret: String,options:NSDictionary? = [:])
```

```swift
let sdkOptions: NSDictionary = ["enableService":"1","servcie":"自定义客服地址"]
ReaderColSDK.shared.config(appid: "1209", secret: "asdf!@", options: sdkOptions)
```

使用

方式一：页面嵌入式接入，例如顶部分栏的子页面

```swift
/// 获取 单页面 (通常当作分页接入，例如顶部分栏的子页面)
@objc public func getHomePageVc()->UIViewController
```

```swift
override func viewDidLoad() {
    super.viewDidLoad()

    let homeVc = ReaderColSDK.shared.getHomePageVc()
    self.addChild(homeVc)
    self.view.addSubview(homeVc.view)
    homeVc.view.snp.makeConstraints { (make) in
        make.edges.equalToSuperview()
    }
}
```

方式二：打开 SDK ， 例如点击 按钮 打开阅读 SDK 页面

```swift
/// 打开 SDK 主页 (tabVC)
@objc public func openColSdkFromViewController(_ controller:UIViewController)

/// 打开书籍详情页
@objc public func openColSdkFromViewController(_ controller:UIViewController,bookId:Int)

/// 打开SDK (target 跳转)
@objc public func openColSdkFromViewController(_ controller:UIViewController,target:String)
```

```swift
class TestViewController: UIViewController {
    @IBAction func btnDidClicked(_ sender: UIButton) {
        ReaderColSDK.shared.openColSdkFromViewController(self)
    }
}
```

广告 ad delegate

```
@objc public protocol ReaderAdDelegate: NSObjectProtocol {

    /// 展示视频广告
    @objc optional func showRewardAd(from viewController: UIViewController,
                                     parameter:NSDictionary,
                                     completion: @escaping ((Bool)->Void))

    /// 每日首次打开阅读器弹窗广告
    @objc optional func newDayFirstOpenReaderAdView(from viewController: UIViewController,
                                                    parameter:NSDictionary,
                                                    renderSuccess:@escaping(_ size:CGSize)->Void) ->  UIView

    /// 章节内容广告
    @objc optional func readerContentAdView(from viewController: UIViewController,
                                            parameter:NSDictionary) ->  UIView

    /// 书籍末页广告
    @objc optional func readerEndAdView(from viewController: UIViewController, parameter:NSDictionary,
                                        renderSuccess:@escaping(_ size:CGSize)->Void) ->  UIView

    /// 信息流广告
    @objc optional func cellAdView(from viewController: UIViewController,parameter:NSDictionary?,
                                   renderSuccess:@escaping(_ size:CGSize)->Void) ->  UIView

}
```

配置代理

```
ReaderColSDK.shared.readerAdDelegate = self
```

实现广告协议

```
// 每日首次打开阅读器弹窗广告 可选 未实现该代理则不会展示每日首次打开阅读器弹窗广告相关
func newDayFirstOpenReaderAdView(from viewController:
    UIViewController,parameter:NSDictionary,renderSuccess:@escaping(_ size:CGSize)->Void) ->  UIView{
    return cellAdView(from: viewController, parameter: parameter, renderSuccess: renderSuccess)
}
```

```
// 激励视频广告 可选 未实现该代理则不会展示激励视频广告相关
func showRewardAd(from viewController: UIViewController,parameter:NSDictionary,completion:@escaping
    ((Bool)->Void)) {
    rewardedVideoAdManager = RewardedVideoAdManager.showVideoAd(with: viewController, videoAdComplation:
        {[weak self] (result) in
        completion(result)
        self?.rewardedVideoAdManager = nil
    })
}
```

```
// 内容中插播广告 可选 未实现该代理则不会展示章节内容间的广告相关
func readerContentAdView(from viewController: UIViewController, parameter:NSDictionary) ->  UIView {
    let adCarrierView = NativeExpressAdCarrierView()
    // 阅读页背面判断
    if let isBackSide =  parameter["isBackSide"] as? Bool , isBackSide == true {
        return adCarrierView
    }
    adCarrierView.loadAd(fromController: viewController, needPlaceHolder: false, centerShow: true,
        renderSuccess: {_ in})
    return adCarrierView
}
```

```
// 阅读末页广告 可选 未实现该代理则不会展示阅读末页广告相关
func readerEndAdView(from viewController: UIViewController,parameter:NSDictionary,renderSuccess:@escaping(_
    size:CGSize)->Void) ->  UIView{
    return cellAdView(from: viewController, parameter: parameter, renderSuccess: renderSuccess)
}
```

```
//信息流广告 可选 未实现该代理则不会展示信息流广告广告相关
func cellAdView(from viewController: UIViewController,parameter:NSDictionary?,renderSuccess:@escaping(_
    size:CGSize)->Void) ->  UIView {
    let adCarrierView = NativeExpressAdCarrierView()
    adCarrierView.loadAd(fromController: viewController,needPlaceHolder: false,centerShow:
        false,renderSuccess: renderSuccess)
    return adCarrierView
}
```