

For easy installation SDK as library use next steps:

1. Add cross-project reference by dragging **jasperserver-mobile-sdk-ios.xcodeproj** file to your application in Xcode
2. Open target settings editor -> **Build Phases** for the target you want to link library into
3. In the tab **Target Dependencies** add dependency on the **jasperserver-mobile-sdk-ios** aggregate target. Target **jasperserver-mobile-ASIHTTPRequest-ios** use inside SDK for including **ASIHTTPRequest** library into SDK (cause of dependency). If you have already installed **ASIHTTPRequest** library (or want to install it [manually](#)) you can disable or remove this target. To do that use next steps in section **Disable including ASIHTTPRequest library for SDK**
4. In the tab **Link Binary With Libraries** link against next frameworks and libraries:
 1. **jasperserver-mobile-sdk-ios.a**
 2. **CFNetwork**
 3. **SystemConfiguration**
 4. **MobileCoreService**
 5. **libxml2.dylib**
 6. **libz.dylib**
5. Open **Build Settings** and add “**/usr/include/libxml2**” to the **Header Search Path** section
6. Use **#import <jasperserver-mobile-sdk-ios/JSCClient.h>** in your source files
7. Build the project to verify installation is successful

For installing SDK as sources:

1. Copy all files from **Classes** directory to your Xcode project
2. SDK require **ASIHTTPRequest** library which locates in **Other Classes** directory. So you can copy all classes from directory or install **ASIHTTPRequest** [manually](#)
3. Open target settings editor -> **Build Phases** for the target you want to link library into
4. In the tab **Link Binary With Libraries** link against next frameworks and libraries:
 1. **CFNetwork**
 2. **SystemConfiguration**
 3. **MobileCoreService**
 4. **libxml2.dylib**
 5. **libz.dylib**
5. Open **Build Settings** and add “**/usr/include/libxml2**” to the **Header Search Path** section
6. Use **#import “JSCClient.h”** in your source files
7. Build the project to verify installation is successful

Disable including ASIHTTPRequest library for SDK:

1. Open **jasperserver-mobile-sdk-ios** target settings
2. Go to **Build Phases** tab
3. In section **Target Dependencies** remove **jasperserver-mobile-ASIHTTPRequest-ios** target
4. In section **Link Binary With Libraries** remove **jasperserver-mobile-ASIHTTPRequest-ios.a** library file

This will build SDK without including **ASIHTTPRequest** directly into library file.