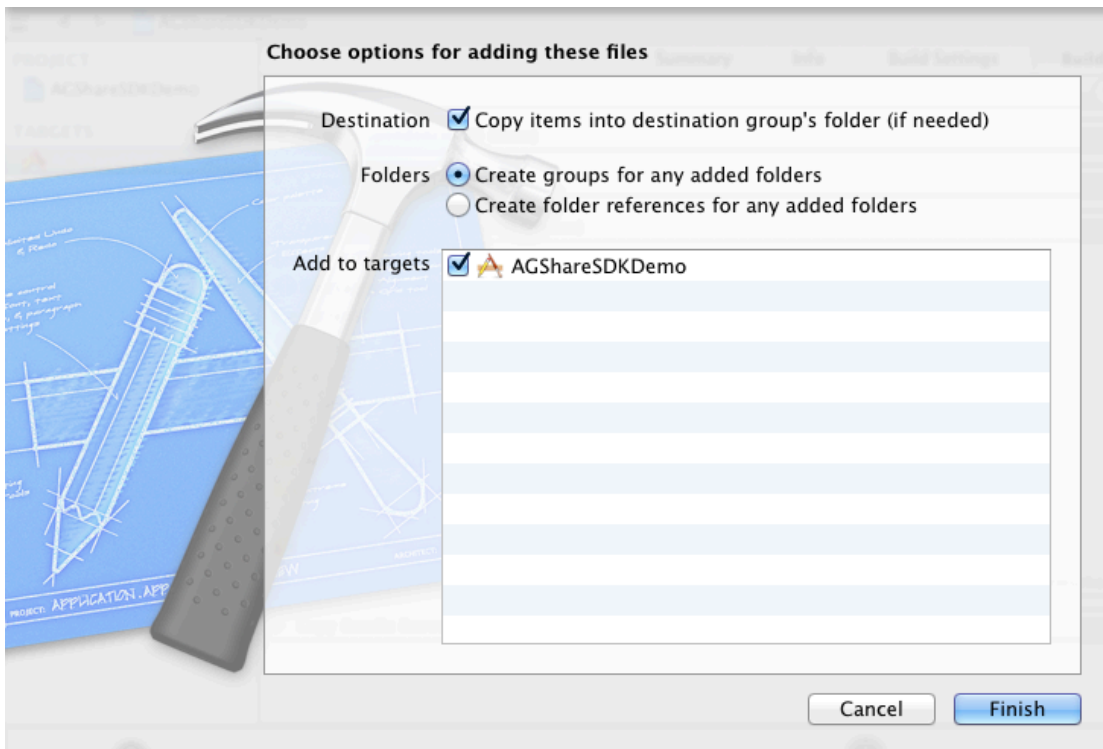


ShareSDK 集成指南

1. 导入 SDK

将 ShareSDK 文件夹到项目文件夹中夹并拖入项目中。



2. 添加依赖框架(Framework)

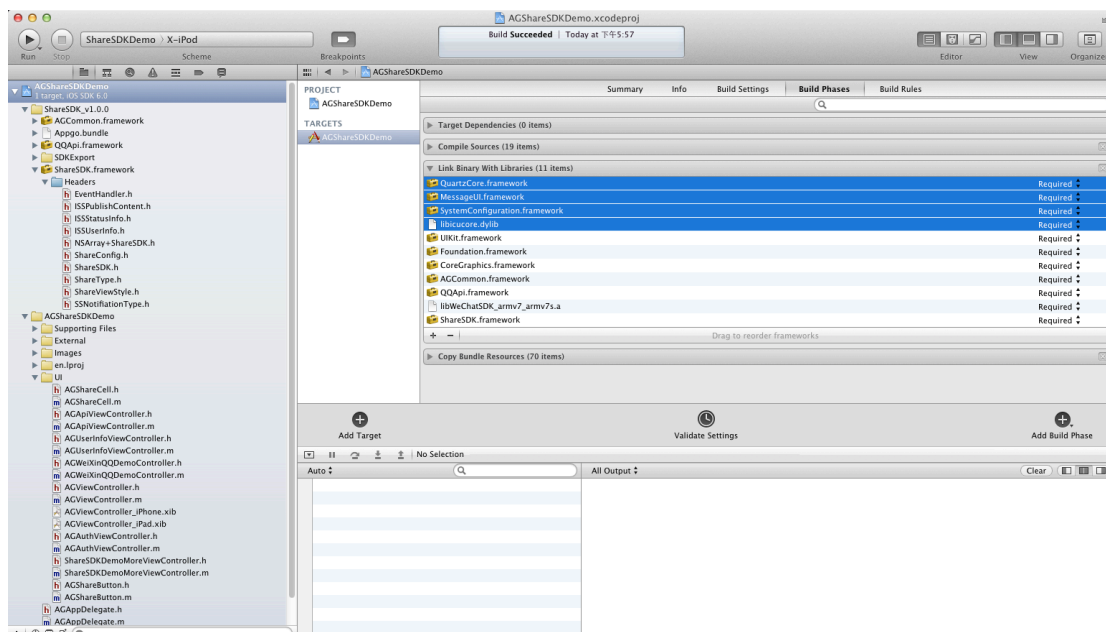
打开项目设置中的 Build Phases 页，在“Link Binary With Libraries”一栏中，点击左下角的“+”号；在弹出窗口里面分别以下库加入到项目中：

SystemConfiguration.framework

QuartzCore.framework

MessageUI.framework

libcucore.dylib



3. 引入社区应用配置信息。

打开 main.m 文件加入

```
#import <ShareSDK/ShareConfig.h>
```

如图：

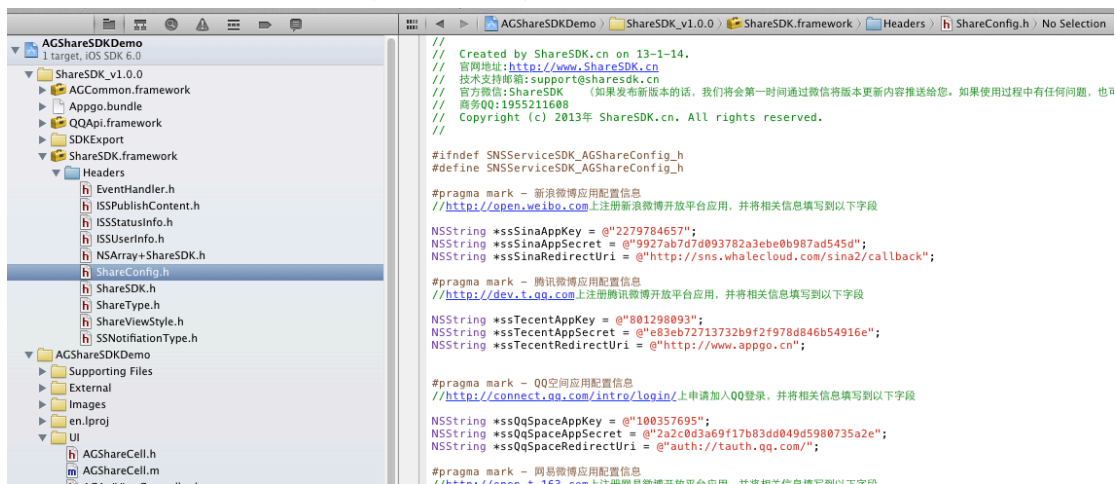
```
//
#import <UIKit/UIKit.h>
#import <ShareSDK/ShareConfig.h>
#import "AGAppDelegate.h"

int main(int argc, char *argv[])
{
    @autoreleasepool {
        return UIApplicationMain(argc, argv, nil, NSStringFromClass([AGAppDelegate class]));
    }
}
```

(P.S. 此头文件必须在此文件下导入，否则会导致编译不通过)

4. 配置所有社交平台的 AppKey

打开 ShareSDKConfig.h 文件，根据需求设置各个平台的 App 相关信息（每个平台的 App 都需要到相应平台上进行应用登记后来取的相关信息）。如图：



5. 配置 URL Scheme

打开*-Info.plist (*代表你的工程名字)。在配置文件中新增一项 URL types

（如果存在可以不创建），展开 URL types – URL Schemes，在 URL Schemes 下新增一项用于新浪微博授权的 Scheme（如果不添加则会导致新浪微博的 SSO 授权方式无法返回应用）。其填写格式为：sinaweibosso.2279784657，其中后面的数字段为你在新浪微博申请的 AppKey。如下图所示：

▼ URL types	Array	(1 item)
▼ Item 0 (Editor)	Dictionary	(3 items)
Document Role	String	Editor
▼ URL Schemes	Array	(3 items)
Item 0	String	sinaweibosso.2279784657
Item 1	String	wx6dd7a9b94f3dd72a
Item 2	String	QQ075BCD15
URL identifier	String	

另外，如果需要使用微信或者 QQ 平台接口也需要在此项下面添加相应的 Scheme。

6. 嵌入代码

打开AppDelegate.m(*代表你的工程名字) 导入文件头 ShareSDK.h:

```
#import <ShareSDK/ShareSDK.h>
```

在 `-(BOOL)application:(UIApplication *)application`

`didFinishLaunchingWithOptions:(NSDictionary *)launchOptions` 方法内添加如下

语句:

```
[ShareSDK registerApp:@"520520test"];
```

然后, 在处理请求 URL 的委托方法中加入 ShareSDK 的处理方法, 如下:

```
- (BOOL)application:(UIApplication *)application handleOpenURL:(NSURL *)url
{
    return [ShareSDK handleOpenURL:url wxDelegate:self];
}
```

```
- (BOOL)application:(UIApplication *)application openURL:(NSURL *)url
sourceApplication:(NSString *)sourceApplication annotation:(id)annotation
{
    return [ShareSDK handleOpenURL:url wxDelegate:self];
}
```

此处理方法已包括微信和 QQ 的回复处理, 如果使用 ShareSDK 内置提示功能可以不需要再实现消息的微信和 QQ 的回复消息捕获。

7. 分享内容

在分享内容部分 ShareSDK 提供了三种接口来满足不同需求的分享功能实现。

7.1 菜单方式分享:

使用此方式进行分享会首先弹出菜单供用户选择分享的目标平台, 然后再显示内容编辑界面供用户进行分享内容编辑, 最后进行分享。调用该方式的接口如下:

```
id<ISSPublishContent> publishContent = [ShareSDK publishContent:@"content"
                                     defaultContent:@""
                                     image:[UIImage imageNamed:@"Icon.png"]
                                     imageQuality:0.8
                                     mediaType:SSPublishContentMediaTypeNews
                                     title:@"ShareSDK"
                                     url:url
                                     musicFileUrl:nil
                                     extInfo:nil
                                     fileData:nil];

[ShareSDK showShareActionSheet:self
                 shareList:shareList
                 content:publishContent]
```

```

        statusBarTips:YES
        oneKeyShareList:[NSArray defaultCenterOneKeyShareList]
        autoAuth:YES
        convertUrl:YES
        shareViewStyle:style
        shareViewTitle:@"内容分享"
        result:^(ShareType type, SSPublishContentState state, id<ISSStatusInfo>
statusInfo, id<ICMErrorInfo> error, BOOL end) {
            if (state == SSPublishContentStateSuccess)
            {
                NSLog(@"成功!");
            }
            else if(state == SSPublishContentStateFail)
            {
                NSLog(@"失败!");
            }
        }
    }];

```

第一个参数用于指定菜单显示在哪个视图上。

第二个参数则用于指定菜单弹出的分享平台列表，传入 `nil` 则表示显示所有分享平台，如果开发者需要自己定制显示列表则通过此参数进行控制。ShareSDK 提供了 `getShareListWithType` 方法来进行分享列表的构造。

第三个参数为实现了 `ISSPublishContent` 协议的对象，需要调用了 ShareSDK 的 `publishContent` 来构造对象。需要特别说明一下的是 `content` 参数对象除了基础的图片和文字字段外还包括了微信和 QQ 这两个平台的其他分享类型参数字段（包括 `News`、`Music`、`Video` 等，通过 `mediaType` 来指定分享出去的类型），如果不需要使用微信和 QQ 的分享功能，可以在构造分享内容对象时把参数值设置为 `nil` 或者默认值即可。

从 1.1.0 版本后可以针对微信、QQ、邮件、短信四种方式进行分享内容定制，`ISSPublishContent` 协议中新增 5 个接口用于单独指定微信好友、微信朋友圈、QQ、邮件、短信所需要的分享内容参数。如下图所示（详细请参考 Demo）：



分享菜单视图



编辑分享内容视图

7.2 分享内容视图方式

使用此方式直接弹出内容分享编辑框，用户编辑完内容后点击发布直接分享出去。其调用代码如下：

```
[ShareSDK shareContentWithType:ShareTypeSinaWeibo
        content:[ShareSDK publishContent: @"content"
        defaultContent:@""]
        image:[UIImage imageNamed: @"Icon.png"]
        imageQuality:0.8
        mediaType:SSPublishContentMediaTypeText]

containerController:self
        statusBarTips:YES
        oneKeyShareList:[NSArray defaultOneKeyShareList]
```



```

shareViewStyle:ShareViewStyleDefault
shareViewTitle:@"内容分享"
result:nil];

```

第一个参数为分享的目标平台类型。其他参数与菜单方式中的 `content` 参数含义相同。其显示效果如上面编辑分享内容视图截图。

7.3 直接分享方式

此方式不需要通过任何分享界面直接发送到指定的平台。调用代码如下：

```

[ShareSDK shareContentWithShareList:[ShareSDK getShareListWithType:ShareTypeSinaWeibo,nil]
content:[ShareSDK publishContent: @"content"
defaultContent:@""
image: [UIImage imageNamed: @"Icon.png"]
imageQuality:0.8
mediaType:SSPublishContentMediaTypeText]
statusBarTips:YES
result:nil];

```

其中第一个参数为分享的目标平台类型数组，使用 ShareSDK 的 `getShareListWithType` 方法进行数组列表构造。其它参数与上面所说的分享接口同名参数含义相同。

7.4 离线分享内容提交。

对于分享失败或者尚未进行分享的内容可以通过调用 ShareSDK 的 `flushOfflineShare` 方法进行离线内容的重新提交。其调用方式如下：

```

[ShareSDK flushOfflineShare:YES result:nil];

```

第一个参数为是否在状态栏显示分享提示。

第二个参数则为分享返回的处理方法。

8. 获取当前用户信息

获取当前授权用户信息，其调用方式如下：

```

[ShareSDK getUserInfoWithType:ShareTypeSinaWeibo
result:^(BOOL result, id<ISSUserInfo> userInfo, id<ICMErrorInfo> error) {
    if (result)
    {
        NSLog(@"成功");
    }
    else
    {
        NSLog(@"失败");
    }
}];

```

其中第一个参数为平台类型，用于指定获取哪个平台的授权用户信息。如果指定平台的用户尚未进行授权则弹出授权界面。第二个参数为授权返回的处理方法。

9. 关注用户

关注指定用户（此接口目前仅支持新浪微博和腾讯微博），其调用方式如下：

```

[ShareSDK followUserWithName:@"ShareSDK"

```

```

shareType:ShareTypeSinaWeibo
result:^(BOOL result, id<ISSUserInfo> userInfo, id<ICErrorInfo> error) {
    if (result)
    {
        NSLog(@"成功");
    }
    else
    {
        NSLog(@"失败");
    }
}
};

```

其中第一个参数为关注用户的昵称，第二个参数为平台类型，用于指定获取哪个平台的授权用户信息。如果指定平台的用户尚未进行授权则弹出授权界面。第二个参数为授权返回的处理方法。

10. 获取关注列表

获取当前授权用户的关注列表（此接口目前仅支持新浪微博和腾讯微博），其调用方式如下：

```

[ShareSDK getFriendsWithType:ShareTypeSinaWeibo
pageNo:1
result:^(BOOL result, NSArray *users, BOOL hasNext, id<ICErrorInfo> error)
{
    if (result)
    {
        NSLog(@"成功");
    }
    else
    {
        NSLog(@"失败");
    }
}
};

```

其中第一个参数为平台类型，用于指定获取哪个平台的授权用户关注列表，如果指定平台的用户尚未进行授权则弹出授权界面。第二个参数为取关注列表的第几页数据，页码从 1 开始。第三个参数为取关注列表返回的处理方法。

11. 用户授权

要想操作相关平台提供的方法必须先取得用户授权。ShareSDK 提供了单独的授权接口来实现用户的登录授权。其代码如下：

```

[ShareSDK authWithType:ShareTypeSinaWeibo result:^(SSAuthState state, id<ICErrorInfo> error) {
    if (state == SSAuthStateSuccess)
    {
        NSLog(@"成功");
    }
    else if (state == SSAuthStateFail)
    {
        NSLog(@"失败");
    }
}
};

```

第一个参数为需要进行授权的平台类型。第二个参数则是授权返回结果的处理方法。需要注意的是 ShareSDK

中提供的各种需要进行授权的接口（包括分享、获取用户信息等）都已经在内部进行授权检测判断，不需要开发者单独进行授权。

授权界面效果如下所示：



授权界面

如果需要取消授权可以调用 ShareSDK 的 `cancelAuthWithType` 接口，如下：

```
[ShareSDK cancelAuthWithType:ShareTypeSinaWeibo];
```