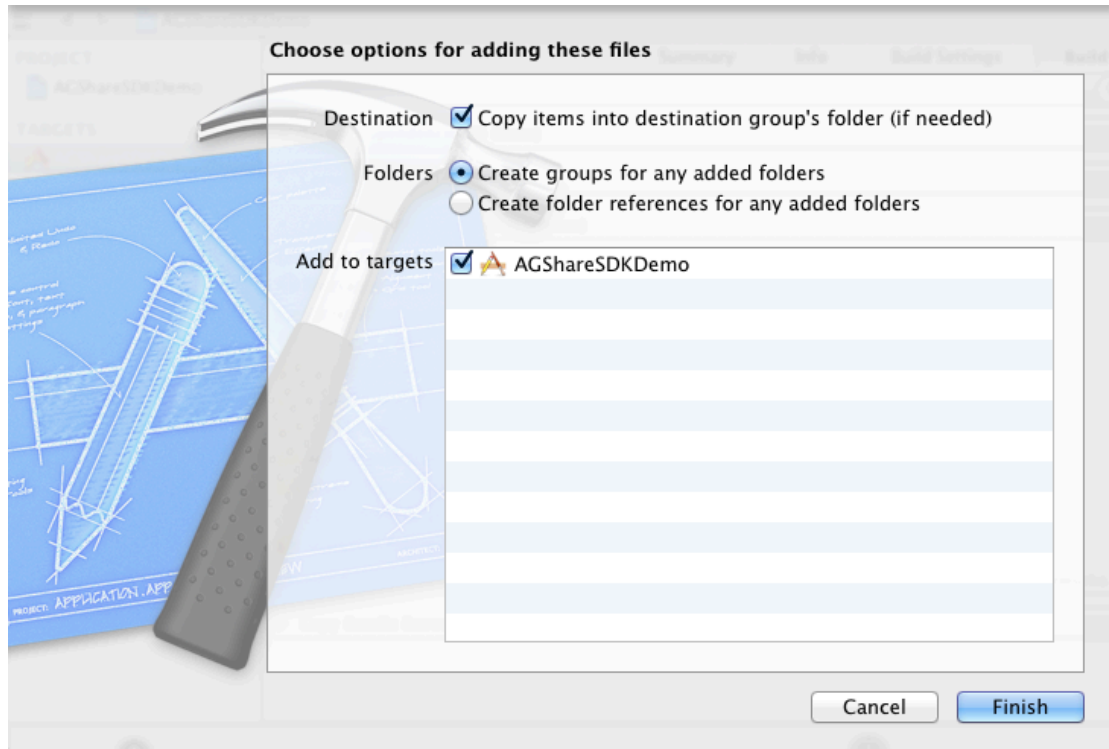


ShareSDK 集成指南

1. 导入 SDK

将 ShareSDK 文件夹到项目文件夹中并拖入项目中。



2. 添加依赖框架(Framework)

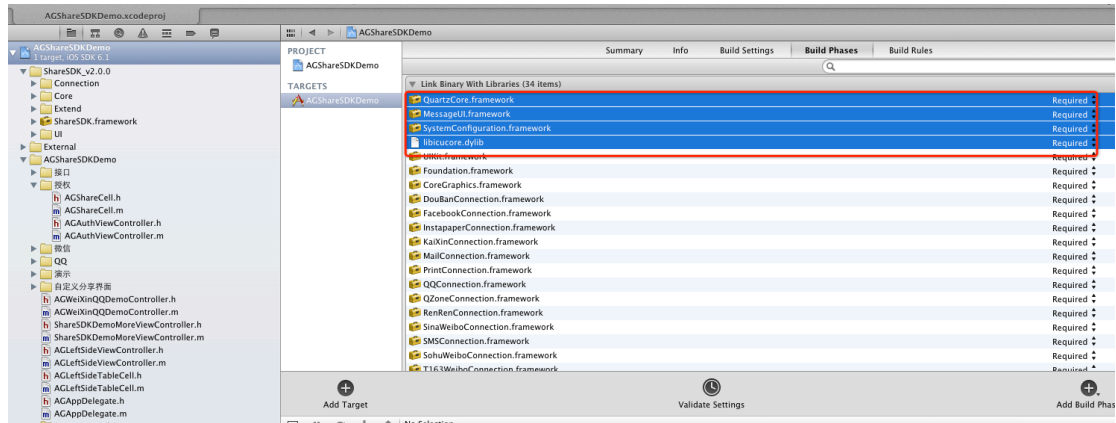
打开项目设置中的 Build Phases 页,在“Link Binary With Libraries”一栏中,点击左下角的“+”号;在弹出窗口里面分别以下库加入到项目中:

SystemConfiguration.framework

QuartzCore.framework

MessageUI.framework (如果不集成邮件和短信可以不添加)

libcucore.dylib



3. 导入头文件 ShareSDK.h 并初始化社交平台的 App 信息。

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: didFinishLaunchingWithOptions:方法中添加如下语句:

```
- (void)initializePlatform
{
    //添加新浪微博应用
    [ShareSDK connectSinaWeiboWithAppKey:@"3201194191"
                                     appSecret:@"0334252914651e8f76bad63337b3b78f"
                                     redirectUri:@"http://appgo.cn"];

    //添加腾讯微博应用
    [ShareSDK connectTencentWeiboWithAppKey:@"801307650"
                                     appSecret:@"ae36f4ee3946e1cbb98d6965b0b2ff5c"
                                     redirectUri:@"http://www.sharesdk.cn"];

    //添加QQ空间应用
    [ShareSDK connectQZoneWithAppKey:@"100371282"
                                     appSecret:@"aed9b0303e3ed1e27bae87c33761161d"];

    //添加网易微博应用
    [ShareSDK connect163WeiboWithAppKey:@"T5EI7BXe13vfyDuy"
                                     appSecret:@"gZxwyN0vjFYpxwlnuizHRRtBRZ2lV1j"
                                     redirectUri:@"http://www.sharesdk.cn"];

    //添加搜狐微博应用
    [ShareSDK connectSohuWeiboWithConsumerKey:@"SAfmTG1bLxZY3HztESWx"
                                     consumerSecret:@"yfTZf)!rVwh*3dqQuVJVsUL37!F)!yS9S!0rcsij"];

    //添加豆瓣应用
    [ShareSDK connectDoubanWithAppKey:@"07d08fbfc1210e931771af3f43632bb9"]
}
```

```

        appSecret:@"e32896161e72be91"
        redirectUri:@"http://dev.kumoway.com/braininference/infos.php"];
//添加人人网应用
[ShareSDK connectRenRenWithAppKey:@"fc5b8aed373c4c27a05b712acba0f8c3"
        appSecret:@"f29df781abdd4f49beca5a2194676ca4"];
//添加开心网应用
[ShareSDK connectKaiXinWithAppKey:@"358443394194887cee81ff5890870c7c"
        appSecret:@"da32179d859c016169f66d90b6db2a23"
        redirectUri:@"http://www.sharesdk.cn/"];
//添加Instapaper应用
[ShareSDK
connectInstapaperWithAppKey:@"4rDJ0Rmc0cSAZL1YpqGHRI605xUvrLb0hkJ07y00wWrYrc
61FA"
        appSecret:@"GNr1Gesp0Qbrm8nvd7rLUsyRQsIo3boIbMguAl9gfpdL0aKZWe"];
//添加有道云笔记应用
[ShareSDK
connectYouDaoNoteWithConsumerKey:@"dcde25dca105bcc36884ed4534dab940"
        consumerSecret:@"d98217b4020e7f1874263795f44838fe"
        redirectUri:@"http://www.sharesdk.cn/"];
//添加Facebook应用
[ShareSDK connectFacebookWithAppKey:@"107704292745179"
        appSecret:@"38053202e1a5fe26c80c753071f0b573"];

//添加Twitter应用
[ShareSDK connectTwitterWithConsumerKey:@"mnTGqtXk0TYMXYTN7qUxg"
        consumerSecret:@"R0kFqr8c3m1HXqS3rm3TJ0WkAJuwB0SaWhPbZ90juc"
        redirectUri:@"http://www.sharesdk.cn"];
}

- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    [ShareSDK registerApp:@"520520test"];
    [self initializePlat];

    .....
    return YES;
}

```

然后，在处理请求 URL 的委托方法中加入 ShareSDK 的处理方法(注：如果你不需要 SSO 登录并且不集成微信、QQ 可省略此步骤)，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
            sourceApplication:sourceApplication
            annotation:annotation
            wxDelegate:self];
}

```

4. 添加微信社交平台（不需要微信分享可略过此步骤）

登录微信开放平台（<http://open.weixin.qq.com/>）注册应用并取得应用的 AppID，然后打开工程中的*AppDelegate.m(*代表你的工程名字) 文件，导入微信 SDK 的头文件：

```
#import "WXApi.h"
```

在- application: didFinishLaunchingWithOptions: 方法内添加如下语句：

```

//添加微信应用
[ShareSDK connectWeChatWithAppId:@"wx6dd7a9b94f3dd72a"
          wechatCls:[WXApi class]];

```

5. 添加 QQ 社交平台（不需要 QQ 好友分享可略过此步骤）

登录手机 QQ 开发者平台（<http://mobile.qq.com/api/>）注册成为开发者，并登记应用取得 APPID，然后打开工程中的*AppDelegate.m(*代表你的工程名字) 文件，导入 QQSDK 的头文件：

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

在- (BOOL)application: didFinishLaunchingWithOptions:方法内添加如下语句：

```

//添加QQ应用
[ShareSDK connectQQWithAppId:@"QQ0F0A941E" qqApiCls:[QQApi class]];

```

6. 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 下分别各新增一项用于新浪微博和 Facebook 授权的 Scheme（**如果不添加则会导致新浪微博或 Facebook 的 SSO 授权方式无法返回应用**）。其中新浪填写格式为：sinaweibosso.2279784657，其中后面的数字段为你在新浪微博申请的 AppKey。Facebook 填写格式为：fb107704292745179，其中后面的数字段为你在 Facebook 申请的 AppKey。如下图所示：

► Required device capabilities	Array	(1 item)
► Supported interface orientations	Array	(1 item)
► Supported interface orientations (iPad)	Array	(1 item)
▼ URL types	Array	(1 item)
▼ Item 0 (Editor)	Dictionary	(3 items)
Document Role	String	Editor
▼ URL Schemes	Array	(4 items)
Item 0	String	fb107704292745179
Item 1	String	sinaweibosso.3201194191
Item 2	String	wx6dd7a9b94f3dd72a
Item 3	String	QQ0758CD15
URL identifier	String	
Icon already includes gloss effects	Boolean	YES
Status bar style	String	Opaque black style

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。

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：

```
NSString *imagePath = [[NSBundle mainBundle] pathForResource:@"ShareSDK"
                                                         ofType:@"jpg"];

//构造分享内容
id<ISSContent> publishContent = [ShareSDK content:@"分享内容"
                                              defaultContent:@"默认分享内容，没内容时显示"
                                              image:[ShareSDK imageWithPath:imagePath]]
```

```

        title:@"ShareSDK"
        url:@"http://www.sharesdk.cn"
        description:@"这是一条测试信息"
        mediaType:SSPublishContentMediaTypeNews];

[ShareSDK showShareActionSheet:nil
        shareList:nil
        content:publishContent
        statusBarTips:YES
        authOptions:nil
        shareOptions: nil
        result:^(ShareType type, SSPublishContentState state,
id<ISSStatusInfo> statusInfo, id<ICMErrorInfo> error, BOOL end) {
            if (state == SSPublishContentStateSuccess)
            {
                NSLog(@"分享成功");
            }
            else if (state == SSPublishContentStateFail)
            {
                NSLog(@"分享失败, 错误码:%d, 错误描述:%@", [error
errorCode], [error errorDescription]);
            }
        }
    }];

```

以上代码中主要使用[ShareSDK showShareActionSheet: shareList: content: statusBarTips: authOptions: shareOptions: esult]方法进行分享。

第一个参数用于指定菜单显示在哪个视图控制器上。如果指定为 nil，则表示使用显示在最顶层的 UIViewController 对象。（注：如果需要适配 iPad 版本，需要调用 container 对象中的相关方法设置 iPad 容器，否则无法在 iPad 中正常显示。）

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 ISSContent 协议的对象，需要调用了 ShareSDK 的+(id<ISSContent>)content: defaultContent: image: title: url: description: mediaType:;方法来构造对象。（注：id<ISSContent>对象提供了对部分平台，如：微信、QQ、QQ 空间、邮件、短信、有道云笔记、Instapaper 平台的内容定制,通过定制可以针对不同的平台发送不同的内容。详细请参考 Demo 程序）

第四个参数为是否在状态栏上显示提示信息（包括成功和错误提示）。

第五个参数用于指定在分享过程中，需要授权时的一些信息配置。如：是否需要自

动弹出授权界面然后授权、授权时是否回调服务器以及授权视图的委托等。需要说明的是在设置是否需要自动授权标志为 **NO** 时，则 **SDK** 对于需要授权的状态不会弹出授权界面，需要用户自行处理。

第六个参数用于指定分享时的一些信息配置。如：分享视图标题、一键分享列表的定制、微信、QQ 分享按钮的隐藏以及分享视图委托等。

第七个参数用于处理分享返回后续的处理方法。返回方法中包括了分享状态、分享成功后的状态信息 (**StatusInfo**)、分享失败时的错误信息、以及本次分享会话是否结束标识。

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：

```
//创建分享内容
NSString *imagePath = [[NSBundle mainBundle] pathForResource:@"image"
ofType:@"jpg"];
id<ISSContent> publishContent = [ShareSDK content:@"content"
                                defaultContent:@""
                                image:[ShareSDK imageWithPath:imagePath]
                                title:nil
                                url:nil
                                description:nil
                                mediaType:SSPublishContentMediaTypeText];

//创建弹出菜单容器
id<ISSContainer> container = [ShareSDK container];
[container setIPadContainerWithView:sender
                                arrowDirect:UIPopoverArrowDirectionUp];

//显示分享菜单
[ShareSDK showShareViewWithType:ShareTypeSinaWeibo
                                container:container
                                content:publishContent
                                statusBarTips:YES
                                authOptions:nil
                                shareOptions:nil
                                result:^(ShareType type, SSPublishContentState state,
id<ISSStatusInfo> statusInfo, id<ICMErrorInfo> error, BOOL end) {
    if (state == SSPublishContentStateSuccess)
    {
        NSLog(@"发表成功");
    }
}
```

```

    }
    else if (state == SSPublishContentStateFail)
    {
        NSLog(@"发布失败!error code == %d, error code
== %@", [error errorCode], [error errorDescription]);
    }
}
}];

```

以上代码主要使用主要使用[ShareSDK showShareViewWithType: container: content: statusBarTips: authOptions: shareOptions: result:]方法进行分享。

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。其提供了两种不同的方法来满足不同的功能，第一种分享到单个平台，调用代码如下：

```

//创建分享内容
NSString *imagePath = [[NSBundle mainBundle] pathForResource:@"image"
ofType:@"jpg"];
id<ISSContent> publishContent = [ShareSDK content:@"content"
defaultContent:@""
image:[ShareSDK imageWithPath:imagePath]
title:nil
url:nil
description:nil
mediaType:SSPublishContentMediaTypeText];

[ShareSDK shareContent:publishContent
type:ShareTypeSinaWeibo
authOptions:nil
statusBarTips:YES
result:^(ShareType type, SSPublishContentState state,
id<ISSStatusInfo> statusInfo, id<ICMErrorInfo> error, BOOL end) {
    if (state == SSPublishContentStateSuccess)
    {
        NSLog(@"分享成功");
    }
    else if (state == SSPublishContentStateFail)
    {
        NSLog(@"分享失败, 错误码:%d, 错误描述:%@", [error errorCode], [error

```



```
errorDescription]);  
    }  
}];
```

以上代码中主要使用[ShareSDK shareContent: type: authOptions: statusBarTips: result:]方法进行分享。其参数说明与上面带分享界面的功能接口参数说明相同。

另外一种方法就是可以通过传入多个平台类型来进行一键分享信息。其调用代码如下：

```
//创建分享内容  
NSString *imagePath = [[NSBundle mainBundle] pathForResource:@"image"  
ofType:@"jpg"];  
id<ISSContent> publishContent = [ShareSDK content:@"content"  
defaultContent:@""  
image:[ShareSDK imageWithPath:imagePath]  
title:nil  
url:nil  
description:nil  
mediaType:SSPublishContentMediaTypeText];  
  
[ShareSDK oneKeyShareContent:publishContent  
shareList:[NSArray defaultOneKeyShareList]  
authOptions:nil  
statusBarTips:nil  
result:^(ShareType type, SSPublishContentState state,  
id<ISSStatusInfo> statusInfo, id<ICMErrorInfo> error, BOOL end) {  
    if (state == SSPublishContentStateSuccess)  
    {  
        NSLog(@"分享成功");  
    }  
    else if (state == SSPublishContentStateFail)  
    {  
        NSLog(@"分享失败, 错误码:%d, 错误描述:%@", [error  
errorCode], [error errorDescription]);  
    }  
}];
```

以上代码主要使用[ShareSDK oneKeyShareContent: shareList: authOptions: statusBarTips: result:]方法进行分享，其中shareList参数表示需要将内容分享到的目标平台类型列表，[NSArray defaultOneKeyShareList]表示分享到所有支持一键分享的平台中。其余参数与其他分享接口意义相同。

7.4 定制分享

对于需要定义分享视图的样式、标题以及部分属性（如控制默认分享视图中的QQ、微信分享按钮显示）或者调整分享视图时，可以在各个分享接口的 `shareOptions` 参数中进行设定。如：

```
//创建分享内容
NSString *imagePath = [[NSBundle mainBundle] pathForResource:@"image"
ofType:@"jpg"];
id<ISSContent> publishContent = [ShareSDK content:@"content"
defaultContent:@""
image:[ShareSDK imageWithPath:imagePath]
title:nil
url:nil
description:nil
mediaType:SSPublishContentMediaTypeText];

//创建分享选项
id<ISSShareOptions> shareOptions =
    [ShareSDK defaultShareOptionsWithTitle:@"内容分享"
oneKeyShareList:[NSArray defaultOneKeyShareList]
qqButtonHidden:YES
wxSessionButtonHidden:YES
wxTimelineButtonHidden:YES
showKeyboardOnAppear:NO
shareViewDelegate:nil
friendsViewDelegate:nil
picViewerViewDelegate:nil];

//弹出分享菜单
[ShareSDK showShareActionSheet:nil
shareList:nil
content:publishContent
statusBarTips:YES
authOptions:nil
shareOptions:shareOptions
result:^(ShareType type, SSPublishContentState state,
id<ISSStatusInfo> statusInfo, id<ICMErrorInfo> error, BOOL end) {
    if (state == SSPublishContentStateSuccess)
    {
        NSLog(@"分享成功");
    }
    else if (state == SSPublishContentStateFail)
    {
    }
```

```

                                NSLog(@"分享失败,错误码:%d,错误描述:%@", [error
errorCode], [error errorDescription]);
                                }
                                }];

```

ShareSDK提供了三种构造ShareOptions对象的方法，分别对应默认、简单以及应用推荐三种分享视图样式，其分别是：

```

+ defaultShareOptionsWithTitle: oneKeyShareList: qqButtonHidden:
wxSessionButtonHidden: wxTimelineButtonHidden:
showKeyboardOnAppear: shareViewDelegate: friendsViewDelegate:
picViewerViewDelegate;;

```

```

+ simpleShareOptionsWithTitle: shareViewDelegate;;

```

```

+ appRecommendShareOptionsWithTile: shareViewDelegate;;

```

title参数表示了设置分享视图的标题，oneKeyShareList参数控制一键分享列表的显示（在默认分享视图底部显示的平台列表），qqButtonHidden、wxSessionButtonHidden、wxTimelineButtonHidden则分别是控制默认分享视图下的QQ、微信好友、微信朋友圈按钮的显示。showKeyboardOnAppear参数控制分享视图显示时是否弹出键盘。shareViewDelegate参数表示分享视图委托，friendsViewDelegate参数表示好友列表视图委托。picViewerViewDelegate参数表示照片查看视图委托。

7.5 定制属于自己的分享 UI

结合+ shareContent: type: authOptions: statusBarTips: result;;和+ authWithType: options: result;;以及+ hasAuthorizedWithType;;方法来打造属于自己的分享UI。下面代码使用简单的UITextView和UIButton来搭建简单的新浪微博分享界面。代码如下：

```

- (void)viewDidLoad
{
    [super viewDidLoad];

    _contentTextView = [[UITextView alloc] initWithFrame:CGRectMake(0.0, 0.0,
320.0, 200.0)];
    [self.view addSubview:_contentTextView];
    [_contentTextView release];

    UIButton *sendButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
    sendButton.frame = CGRectMake(0.0, 205.0, 100.0, 45.0);
    [sendButton setTitle:@"分享" forState:UIControlStateNormal];

```

```

[sendButton addTarget:self
                action:@selector(sendButtonClickHandler:)
                forControlEvents:UIControlEventTouchUpInside];
[self.view addSubview:sendButton];
}

- (void)sendButtonClickHandler:(id)sender
{
    if ([ShareSDK hasAuthorizedWithType:ShareTypeSinaWeibo])
    {
        //构造分享内容
        id<ISSContent> publishContent =
            [ShareSDK content:_contentTextView.text
                    defaultContent:nil
                    image:nil
                    title:nil
                    url:nil
                    description:nil
                    mediaType:SSPublishContentMediaTypeNews];

        [ShareSDK shareContent:publishContent
                    type:ShareTypeSinaWeibo
                    authOptions:nil
                    statusBarTips:YES
                    result:^(ShareType type, SSPublishContentState state,
id<ISSStatusInfo> statusInfo, id<ICMErrorInfo> error, BOOL end) {
            if (state == SSPublishContentStateSuccess)
            {
                NSLog(@"分享成功");
            }
            else if (state == SSPublishContentStateFail)
            {
                NSLog(@"分享失败, 错误码:%d, 错误描述:%@", [error
errorCode], [error errorDescription]);
            }
        }
    }
}

else
{
    [ShareSDK authWithType:ShareTypeSinaWeibo
                authOptions:nil
                result:^(SSAuthState state, id<ICMErrorInfo> error) {
        if (state == SSPublishContentStateSuccess)
        {

```

```

//构造分享内容
id<ISSContent> publishContent =
    [ShareSDK content:_contentTextView.text
        defaultContent:nil
        image:nil
        title:nil
        url:nil
        description:nil
        mediaType:SSPublishContentMediaTypeNews];

[ShareSDK shareContent:publishContent
    type:ShareTypeSinaWeibo
    authOptions:nil
    statusBarTips:YES
    result:^(ShareType type, SSPublishContentState
state, id<ISSStatusInfo> statusInfo, id<ICMErrorInfo> error, BOOL end) {
        if (state == SSPublishContentStateSuccess)
        {
            NSLog(@"分享成功");
        }
        else if (state == SSPublishContentStateFail)
        {
            NSLog(@"分享失败, 错误码:%d, 错误描述:%@",
[error errorCode], [error errorDescription]);
        }
    }
}];
}
}
}

```

7.6 iPad 版本分享界面集成

如果需要在iPad中显示分享，则需要使用+ container;功能接口创建一个容器对象并调用容器对象的setIphoneContainer系列方法来设置iPad版显示容器。其实现代码如下：

[illegible]

```

//弹出分享菜单
[ShareSDK showShareActionSheet:container
    shareList:nil
    content:publishContent
    statusBarTips:YES
    authOptions:authOptions
    shareOptions:shareOptions
    result:^(ShareType type, SSPublishContentState state,
id<ISSStatusInfo> statusInfo, id<ICMErrorInfo> error, BOOL end) {
    if (state == SSPublishContentStateSuccess)
    {
        NSLog(@"分享成功");
    }
    else if (state == SSPublishContentStateFail)
    {
        NSLog(@"分享失败, 错误码:%d, 错误描述:%@", [error
errorCode], [error errorDescription]);
    }
}
}];

```

8. 激活 / 禁止 SSO 登录。

用于控制平台是否使用SSO登录方式进行登录（目前只支持新浪微博、Facebook），如下代码禁止SSO登录：

```

[ShareSDK ssoEnabled:_ssoEnable];

```

9. 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
    authOptions:nil
    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。第二个参数为授权选项参数。第三个参数为授权返回的处理方法。

10. 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：

```
//关注用户  
[ShareSDK followUserWithType:ShareTypeSinaWeibo  
    field:@"ShareSDK"  
    fieldType:SSUserFieldTypeTypeName  
    authOptions:nil  
    viewDelegate:nil  
    result:^(SSResponseState state, id<ISSUserInfo> userInfo,  
id<ICMErrorInfo> error) {  
    NSString *msg = nil;  
    if (state == SSResponseStateSuccess)  
    {  
        NSLog(@"关注成功");  
    }  
    else if (state == SSResponseStateFail)  
    {  
        NSLog(@"%@ ", [NSString stringWithFormat:@"关  
注失败:%@", error.errorDescription]);  
    }  
    }];
```

其中第一个参数为关注用户所属平台类型，第二个参数为关注用户的昵称或者用户ID（具体指示昵称还是用户ID由第三个参数控制），第三个参数为标识第二个参数的字段类型）。第四个参数为授权选项配置，第五个参数为视图委托，主要用于Facebook关注时弹出的添加好友界面的委托。

11 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、腾讯微博、Facebook、Twitter），其调用方式如下：

```
id<ISSPage> page = [ShareSDK pageWithPageNo:1 pageSize:0];
```

```
[ShareSDK getFriendsWithType:ShareTypeSinaWeibo
        page:page
        authOptions:nil
        result:^(BOOL result, NSArray *users, id<ISSPage>
currPage, id<ISSPage> prevPage, id<ISSPage> nextPage, BOOL hasNext,
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分页对象，用于指定要获取那一页的好友列表数据；ShareSDK中提供了两种构造分页对象的方法+ pageWithCursor::和+ pageWithPageNo: pageSize::。第三个参数为授权选项设置，第四个参数为关注列表返回的处理方法。

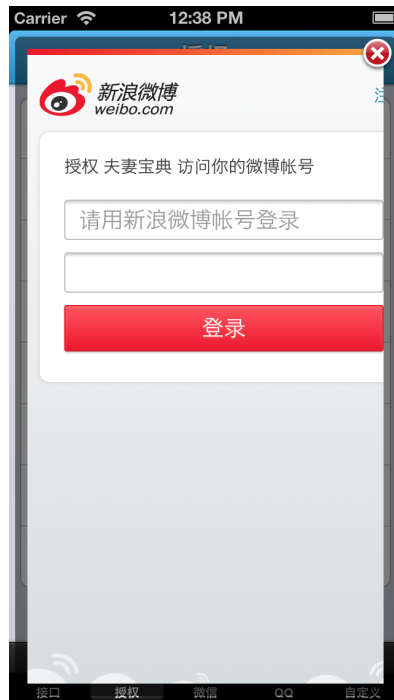
12 用户授权

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 options:nil result:^(SSAuthState
state, id<ICMErrorInfo> 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。第二个参数为授权选项设置，如果需要定制授权视图的样式需要指定此参数。第三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

13. 用户取消授权

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

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

14. 从外部授权信息中设置 ShareSDK 平台授权。

如果代码中已实现某平台的授权，那么可以通过ShareSDK中的+setCredential::;接口设置平台的授权信息。如下：

```
id<ISSCredential> newCredential = [ShareSDK  
credentialWithSourceData:sourceData type:ShareTypeSinaWeibo];  
  
//设置新浪微博使用新的授权凭证  
[ShareSDK setCredential:newCredential type:ShareTypeSinaWeibo];
```

其中sourceData为授权返回的数据。