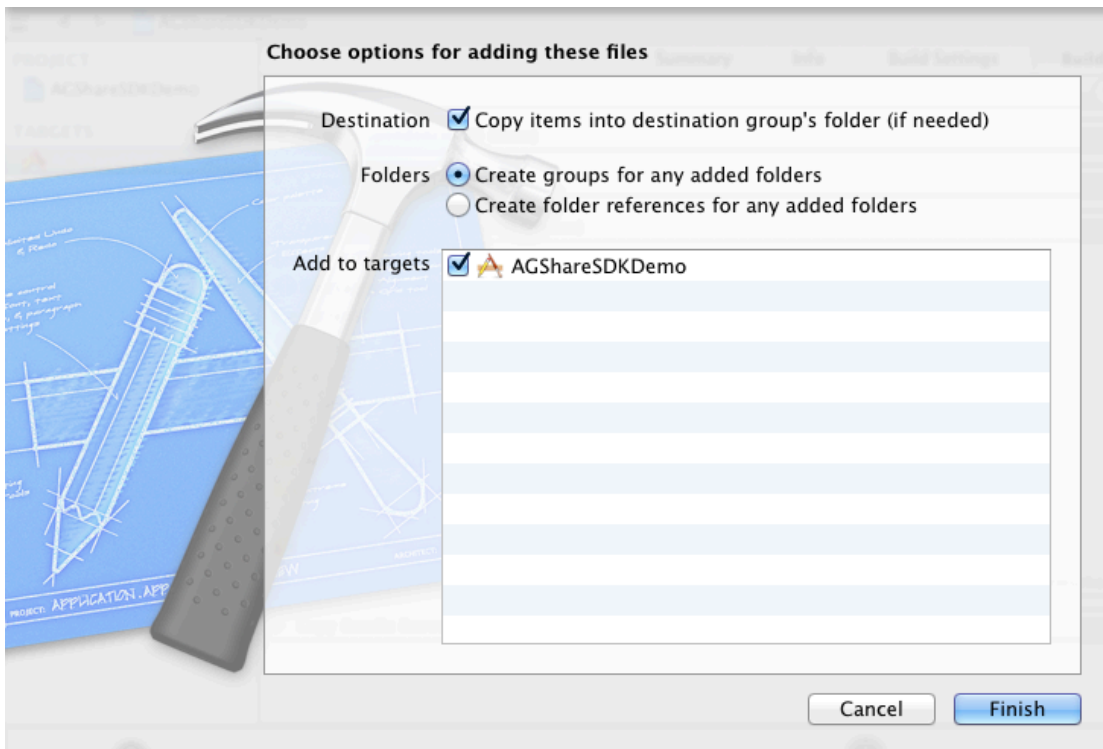


# ShareSDK 集成指南

## 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)

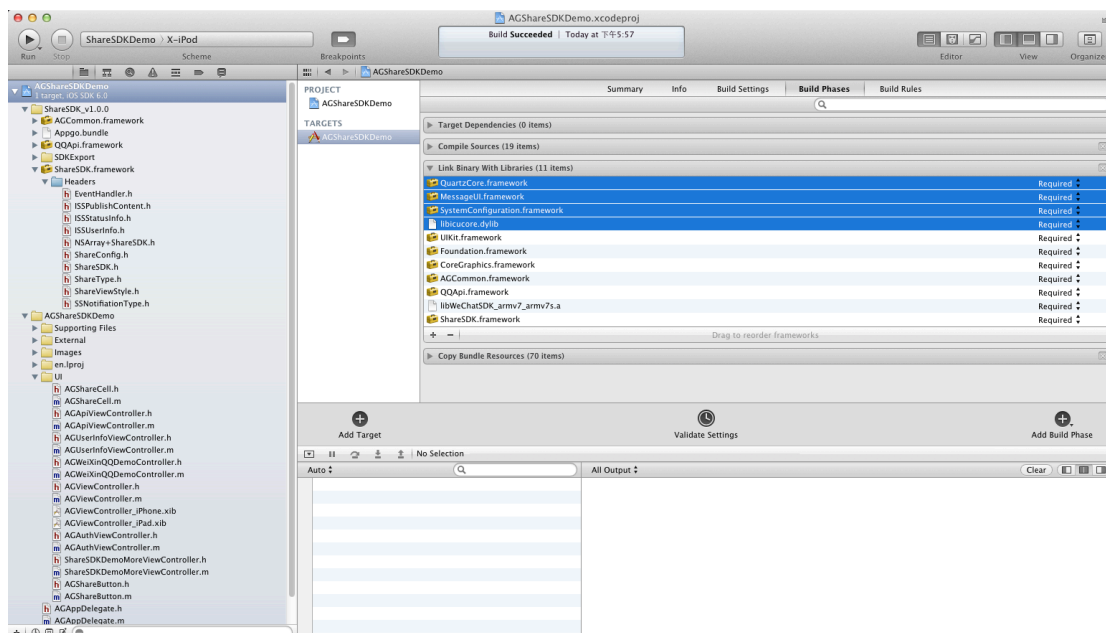
打开项目设置中的 Build Phases 页，在“Link Binary With Libraries”一栏中，点击左下角的“+”号；在弹出窗口里面分别以下库加入到项目中：

SystemConfiguration.framework

QuartzCore.framework

MessageUI.framework

libcure.dylib



### 3. 引入社区应用配置信息。

打开 main.m 文件加入

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

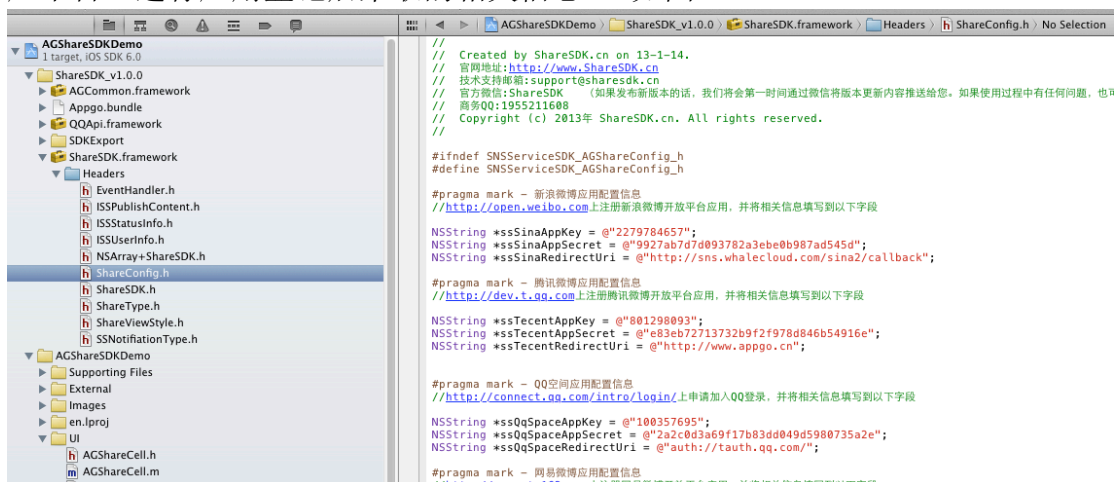
如图：

```
//  
  
#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	QQ075BCD15
URL identifier	String	
Icon already includes gloss effects	Boolean	YES
Custom 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。

## 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"];
```

注: `registerApp` 中传入参数为开发者在 ShareSDK 官方网站申请的 AppKey。

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 defaultOneKeyShareList]
      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）：

```

(void)shareAllButtonClickHandler:(UIButton *)sender
{
    id<ISSPublishContent> publishContent = [ShareSDK publishContent:CONTENT
    defaultContent:@""]
    image:[UIImage imageNamed:IMAGE_NAME]
    imageQuality:0.8
    mediaType:SSPublishContentMediaTypeNews
    title:@"ShareSDK"
    url:@"http://www.sharesdk.cn"
    musicFileUrl:nil
    extInfo:nil
    fileData:nil];

    //定制微信好友内容
    [publishContent addWeixinSessionUnitWithType:INHERIT_VALUE
    content:@"Hello 微信好友!"
    title:INHERIT_VALUE
    url:INHERIT_VALUE
    image:[UIImage imageNamed:IMAGE_NAME]
    imageQuality:INHERIT_VALUE
    musicFileUrl:INHERIT_VALUE
    extInfo:INHERIT_VALUE
    fileData:INHERIT_VALUE];

    //定制微信朋友圈内容
    [publishContent addWeixinTimelineUnitWithType:[NSNumber numberWithInt:SSPublishContentMediaTypeMusic]
    content:@"Hello 微信朋友圈!"
    title:INHERIT_VALUE
    url:@"http://y.qq.com/i/
    song.html#p=7822736F6E675F4E616D65223A22E4BD0AE48880E698AFE79C9FE6AD3E79A84E5BFABE48990:
    A22687474703A2F2F4736D7573696332342E74632E71712E636F6D2F586B830351563558484645574315071
    5444656377343563753863556728474B304964794E6A3770633447524A574C487953338302F363436323233:
    66469723D423203312663743D382663689643D222C22736F6E675F576966695524C223A22687474703:
    71712E636F6D2F3303634363232326D703322C226E657454797065223A227769666922C22736F6E675F:
    FE5B7A8E89888444E414C495645F8BC1E6BC94E594B1E4BC9AE58996E7BAAA58D095E99BF3222C22736F6E6
    797065223A12C22736F6E675F53696E67657223A22E4BA94E69C88E5A49222C22736F6E675F7170446F:
    4736D757369633132382E74632E71712E636F6D2F586C464E4D31354C5569396961495674593739786D43653:
    4378494C73484D6C6A307849634A454B394568572FE4E39784648316368316F37636848323568413D3D2F3330:
    03738313938266469723D42322666D302663743D382663689643D2673747265616D5F706F733D3827D0"]
    image:[UIImage imageNamed:IMAGE_NAME]
    imageQuality:INHERIT_VALUE
    musicFileUrl:@"http://mp3.mwap8.com/destdir/Music/2009/20090601/ZuiXuanMinZuFeng20090601119.mp3"
    extInfo:nil
    fileData:nil];

    //定制QQ分享内容
    [publishContent addQQUnitWithType:INHERIT_VALUE
    content:@"Hello QQ!"
    title:INHERIT_VALUE
    url:INHERIT_VALUE
    image:[UIImage imageNamed:IMAGE_NAME]
    imageQuality:INHERIT_VALUE];

    //定制邮件分享内容
    [publishContent addMailUnitWithSubject:INHERIT_VALUE
    content:@"<a href='http://sharesdk.cn'>Hello Mail</a>"
    isHTML:[NSNumber numberWithInt:YES]
    attachData:[UIImage imageNamed:IMAGE_NAME]
    imageQuality:INHERIT_VALUE];
}

```

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）。

第六个参数用于指定在分享过程中，如果授权失效是否委托 SDK 进行用户授权。如果为 YES 则表示用户授权过期后自动弹出授权界面让用户授权。否则需要开发者自己判断错误代码然后进行处理。（P.S. 此方法在 v1.0.1 版本中开始支持）

第八个参数用于指定分享内容编辑视图的样式，ShareSDK 提供了三种方式，具体显示样式可以参照下面提供的应用截图。

第十个参数用于处理分享返回后续的处理方法。  
菜单分享方式的效果如下图所示：



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。

### 7.5 设置分享视图属性

v1.1.2 版本中对分享视图的属性设置进行了整合（原有接口保持原有调用方式），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）通过以下方法进行调用：

```

[ShareSDK showShareActionSheet:self
shareList:nil
content:publishContent
statusBarTips:YES
autoAuth:YES
convertUrl:YES
shareViewOptions:[ShareSDK defaultShareViewOptionsWithTitle:@"内容分享"
oneKeyShareList:[NSArray
defaultOneKeyShareList]

qqButtonHidden:YES
wxSessionButtonHidden:YES
wxTimelineButtonHidden:YES]
result:^(ShareType type, SSPublishContentState state, id<ISSStatusInfo>
statusInfo, id<ICMErrorInfo> error, BOOL end) {
if (state == SSPublishContentStateSuccess)
{

```

```

        NSLog(@"发送成功");
    }
    else
    {
        NSLog(@"发送失败");
    }
}];

```

其中的 `shareViewOptions` 参数则是指定分享视图的显示属性，SDK 目前提供了三种构造该参数的方法：

```

+ (id<IShareViewOptions>)defaultShareViewOptionsWithTitle:(NSString *)title
                        oneKeyShareList:(NSArray *)oneKeyShareList
                        qqButtonHidden:(BOOL)qqButtonHidden
                        wxSessionButtonHidden:(BOOL)wxSessionButtonHidden
                        wxTimelineButtonHidden:(BOOL)wxTimelineButtonHidden;
+ (id<IShareViewOptions>)simpleShareViewOptionWithTitle:(NSString *)title;
+ (id<IShareViewOptions>)appRecommendShareViewOptionWithTitle:(NSString *)title;

```

分别对应默认视图样式、简约视图样式、应用推荐样式的属性对象构造。

## 7.6 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（目前只有新浪微博支持），如下代码禁止 SSO 登录：

```

[ShareSDK ssoEnabled:_ssoEnable];

```

## 7.7 定制属于自己的分享 UI

结合

```

+ (void)shareContentWithShareList:(NSArray *)shareList
                        content:(id<ISSPublishContent>)content
                        statusBarTips:(BOOL)statusBarTips
                        result:(PublishContentEventHandler)result;
+ (void)authWithType:(ShareType)type result:(AuthEventHandler)result;

```

和

```

+ (BOOL)hasAuthorizedWithType:(ShareType)type;

```

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];

    // Do any additional setup after loading the view, typically from a nib.

    _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])
    {
        [ShareSDK shareContentWithShareList:[ShareSDK getShareListWithType:ShareTypeSinaWeibo,nil]
            content:[ShareSDK publishContent:_contentTextView.text
                defaultContent:nil
                image:nil
                imageQuality:0
                mediaType:SSPublishContentMediaTypeText]
            statusBarTips:YES
            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!");
                }
            }
        ]];
    }
    else
    {
        [ShareSDK authWithType:ShareTypeSinaWeibo result:^(SSAuthState state, id<ICMErrorInfo> error)
        {
            if (state == SSPublishContentStateSuccess)
            {
                [ShareSDK shareContentWithShareList:[ShareSDK
getShareListWithType:ShareTypeSinaWeibo,nil]
                    content:[ShareSDK publishContent:_contentTextView.text
                        defaultContent:nil
                        image:nil
                        imageQuality:0
                        mediaType:SSPublishContentMediaTypeText]
                    statusBarTips:YES
                    result:^(ShareType type, SSPublishContentState state,
id<ISSStatusInfo> statusInfo, id<ICMErrorInfo> error, BOOL end) {
                        if (state == SSPublishContentStateSuccess)
                        {
                            NSLog(@"成功!");
                        }
                    }
                ]];
            }
        }];
    }
}

```



```

        pageNo:1
        result:^(BOOL result, NSArray *users, 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：

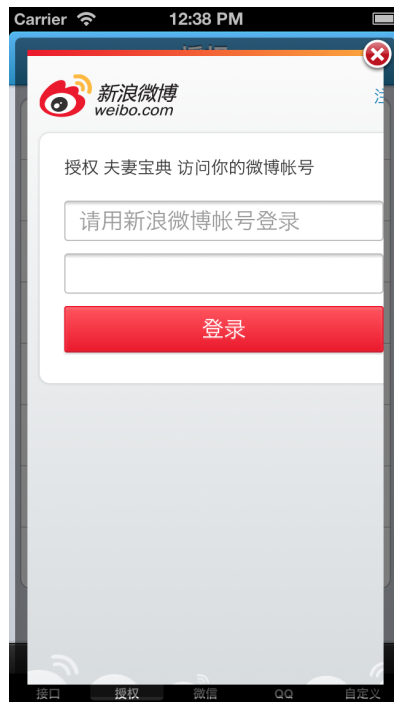
```

[ShareSDK authWithType:ShareTypeSinaWeibo result:^(SSAuthState state, id<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

## 12. 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];
```

## 13. 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 `setClientToken` 接口设置平台的授权信息。如下：

```
id<IClientToken> token = [ShareSDK clientTokenWithData:data type:ShareTypeSinaWeibo];  
[ShareSDK setClientToken:token];
```

其中 `data` 为授权返回的数据。