

1.快速集成指南

1.1 产品概述

友盟社会化分享服务帮助你为你的用户提供社会化分享功能，连接用户和他的社会关系，方便他们分享在你的应用中的精彩体验。

1. 评论,用户可以使用自己的微博账号或者匿名评论到应用自己的评论列表中。用户输入评论时可以选择分享自己的地理位置信息。同时评论时可以选择分享到任意微博平台。
2. 转发到微博，用户可以转发信息到任意微博。
3. 喜欢，点击喜欢按钮，看到喜欢数量。
4. 用户账户管理，对任意微博平台授权一个账户，绑定任意一个微博账户作为评论账号。

1.2 集成步骤

1.2.1 申请友盟应用标识

如果你之前已经在友盟注册了应用，获得了 `UMENG_APPKEY`，可以继续使用之前获得 `UMENG_APPKEY`。

如果你尚未在友盟注册应用，你可以登陆你的账号，点击添加新应用，完成新应用填写之后，讲进入应用管理页面。在该页面就能得到 `Appkey`。

1.2.2 绑定友盟应用标识与各开放平台

获得appkey后，你还需要将你的应用和我们支持的分享平台注册时获取的appkey和appsecret绑定。如果你尚未在我们支持的分享平台注册你的应用，注册过程可以参考如下：

新浪微博：`http://open.t.sina.com.cn -> 我是开发者 -> 创建新应用`

在应用信息 -> 高级信息下做如下3项设置才能正常显示：

1. 把授权设置下的应用回调页设为：`http://sns.whalecloud.com/sina2/callback`
2. 把安全设置下的“绑定域名”设为：`sns.whalecloud.com`
3. 如果应用尚未通过审核，则需要指定测试用户，该阶段只有指定的测试用户才可以分享信息到新浪微博

腾讯微博：`http://open.t.qq.com/ -> 创建应用`

授权回调地址为：`http://sns.whalecloud.com/tenc2/callback`

人人网：`http://dev.renren.com/ -> 创建应用`

- 1.安全设置->授权回调地址为: `http://sns.whalecloud.com/renr2/callback`
- 2.基本信息->应用根域名: ``sns.whalecloud.com``

豆瓣: `http://developers.douban.com/apikey/->创建应用`

回调地址为: `http://sns.whalecloud.com/douban/callback`

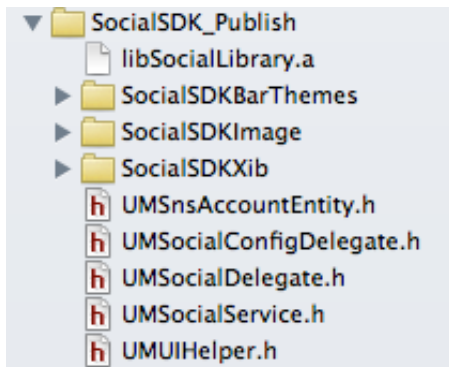
QQ空间 `http://open.qq.com/->创建应用`

选择应用管理->QQ空间平台信息 应用开发地址设置为: `http://sns.whalecloud.com/qzone/c`

在各平台注册你的应用后, 接下来你需要完成UMENG_APPKEY和你在我们支持的分享平台注册你的应用时获取的appkey和appsecret绑定。具体操作为: `开发工具->SNS分享->设置`。

1.2.3 向工程中添加社会化组件安装包

1. 在Finder中, 将 `SocialSDK_Publishment/` 文件夹拖拽到你的工程目录中 (**请注意**: 如果你是升级友盟社会化分享SDK, 请确保在操作前, 彻底删除了旧版本中的所有文件)。
2. 添加到项目中: 将 `SocialSDK_Publishment/` 文件夹, 拖到XCode项目中, 并在弹窗中, 勾选相应的target。



1.2.4 向工程中提那家依赖框架

添加相应Framework:

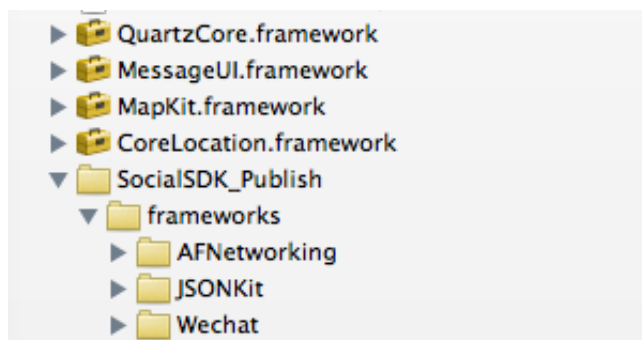
- `MessageUI.Framework`
- `CoreLocation.Framework`
- `MapKit.Framework`
- `QuartzCore.Framework`

本SDK网络交互用到AFNetworking这个网络交互框架, 因为考虑到app中也有可能用到这个框架, 所以没有把AFNetworking的源代码打包到SDK的库里面, 我们已经把它放在 `SocialSDK_Publishment/frameworks/AFNetworking` 里面。注意的是这个框架不支持arc, 如果你的app使用arc的话需要如下操作: 选择项目->Build Phases->Compile Sources, 选

择AFNetworking的文件双击文件在添加编译标志 `-fno-objc-arc`。

本SDK中需要JSON解析库的支持，同样你可以直接使用 `SocialSDK_Publishment/frameworks/JSONKit`，你也可以自己添加其他JSON解析库，例如 `JSONKit`，`SBJSON`，`yajl`，`Nextive JSON`，`NSJSONSerialization`，同AFNetworking一样，`JSONKit`同样不支持arc，如果你的app使用arc，也需要和上面一样的操作。

如果你需要集成微信分享，同样需要把微信的SDK集成进来，我们已经把它放在了 `SocialSDK_Publishment/frameworks/Wechat` 里面，而且你需要参照demo的写法，在分享里面中就出现了微信分享。

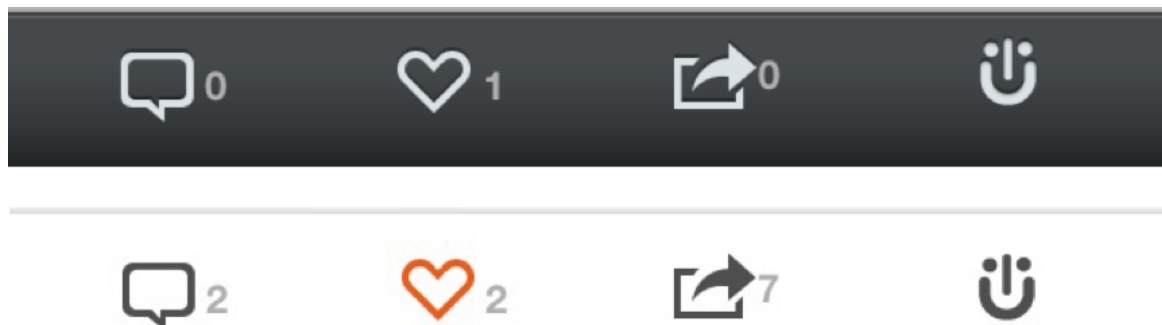


1.2.5 导入组件头文件并配置友盟应用标识

在 `SocialSDK_Publishment` 内有4个头文件，其中 `UMSocialService.h` 是要类的基本头文件，引进头文件之后设置SDK的 `UMENG_APPKEY`，例如下面在 `AppDelegate` 的 `didFinishLaunchingWithOptions` 方法内添加如下代码：`[UMSocialData setAppKey:UMENG_APPKEY];`

1.2.6 把社会化组件加入您的应用

上述步骤已把全部环境准备好，现在来着手将社会化组件加入APP吧！社会化组件的核心组件是“社会化操作栏”，如下图所示：



这是个放置在APP底部的通用工具栏，它为用户提供了针对特定内容页的一站式社会化功能（评论、喜欢、分享、个人中心4个功能）。请注意，如果您需要单独使用某项功能，请您查看

本文档的“开发指南”部分。

注意事项：APP中的每个社会化操作栏实例唯一对应一个Entity。一个Entity其实就是您app中的一个内容项。每个社会化操作（即评论、喜欢、分享、浏览）会对应一个Entity。

一个Entity可以是如下内容项：一个站点、一张照片、一篇新闻，但是它在APP内部“必须”有一个唯一标识Identifier。在实例化社会化操作栏时，创建一个Entity即为创建一个UMSocialData对象，使用用一个字符串对象作为identifier。

使用社会化操作栏非常简单——实例化一个“社会化操作服务”，并将其“社会化操作栏”变量添加到您的视图控制器中即可。代码如下：

```
- (void)viewDidLoad{
    [super viewDidLoad];
    UMSocialData *socialData = [[UMSocialData alloc] initWithIdentifier:@"y
    UMSocialBar *socialBar = [[UMSocialBar alloc] initWithUMSocialData:soci
    [self.view addSubview:socialBar];
}
```

1.2.7相关配置

如果你要集成的微博平台不是我们支持的平台，例如只要新浪微博和腾讯微博的话，可以自己指定在AppDelegate 的 didFinishLaunchingWithOptions 方法内添加如下代码：

[UMSocialControllerService setSocialConfigDelegate:self]; 并且实现 UMSocialConfigDelegate 的 shareToPlatforms 方法

```
- (NSArray *)shareToPlatforms
{
    NSNumber *sinaNumber = [NSNumber numberWithInt:UMShareToTypeSina];
    NSArray *shareToArray = [NSArray arrayWithObjects:sinaNumber, nil];
    return shareToArray;
}
```

同理，如果你要设置其他配置，实现 UMSocialConfigDelegate 的其他方法，具体可以参见 api文档。

1.3 验证集成效果

注意：如果您希望社会化操作栏使用黑色主题，请将组件安装包中的 SocialSDK_Publish/SocialSDKBarThemes/White 这个主题目录删掉。编译运行一下吧，如果您现在能看到这样一个漂亮的“社会化操作栏”，评论、喜欢、分享、个人中心几个按钮功能都正常的话，恭喜您，您的APP已经插上了社交的翅膀！



还有让人兴奋的东西 —— 您可以登录友盟主站，进入组件->社会化分享报表页，这里有您关心的所有社会化操作统计数据。

注意事项：如果您希望单独使用友盟社会化组件提供的功能，或者希望更深层次的功能和UI定制，请移步至下一章 —— “开发指南”部分。

2. 开发指南

2.1 内容 (Entity)

2.1.1 简介

一个Entity就是你APP中的一个内容项。通篇的文档和代码中，我们都会用Entity来指代。这仅仅是个通用术语，来指代程序中可以浏览、分享、评论、喜欢的任何东西。通常它会对应你APP中的一个内容项。这些Entity必须要与一个唯一identifier相对应。如果可能，建议您使用http url来指定Entity的key。

2.1.2 相关方法说明

一个UMSocialData对象标识一个分享资源，用一个identifier字符串作为标识，你可以为此对象设置分享内嵌文字、分享图片等，可以获取到对应的分享数、评论数。初始化方法为 `-(id)initWithIdentifier:(NSString *)identifier`。

2.2 用户

2.2.1 简介

每个使用社会化组件的设备都会对应一个“用户”，但是组件会为您自动处理“用户”对象，您无需显式创建它。用户信息分两部分：登录信息：APP内部展示给其他用户的个人信息。系统会将用户登录时选择的默认平台上的授权信息，作为用户的登录信息；授权信息：从各开放平台上获得的个人资料，如昵称、头像等。

2.2.2 用户中心（界面的接口）

个人中心页面，个人中心页面，上面是登陆的微博账号，下面是各个已经授权的微博账号，包

括我们支持的所有平台，可以对各个平台进行授权和解除授权。



实现的代码是

```
UMSocialData *socialData = [[UMSocialData alloc] initWithIdentifier:@"test"]  
UMSocialControllerService *socialController = [[UMSocialControllerService c  
UINavigationController *accountViewController =[socialController getSocialA  
[self presentViewController:accountViewController animated:YES];
```

2.2.3 获取用户微博账号信息（不带界面的接口）

获取用户微博基本信息的接口

```
//如上对socialController进行初始化  
[socialController.socialDataService requestSocialAccount];
```

并设置实现回调函数的对象

```
[socialController.socialDataService setUMSoicalDelegate:self];
```

在 `-(void)didFinishGetUMSocialDataResponse:(UMSocialResponseEntity *)response` 方法内的 `response` 对象的 `data` 属性可以获取到各个授权的微博账号的基本信息，包括性别、头像、页面url、用户名、用户id。

如果想获取本地缓存的用户信息，可以直接用 `UMSocialData` 的 `socialAccount` 属性，它是一个 `NSDictionary` 对象。键为各个微博平台名称，我们把他们定义好为常量字符串 `UMShareToSina`、`UMShareToTencent`值为 `UMSocialAccountEntity` 对象。

如果想获取各个微博账号详细的信息，可以调用 `UMSocialDataService` 的 `-(void)requestSnsInfomation:(UMShareToType)shareToType;` 方法。同样在回调方法来得到用户微博数据。

2.2.4 授权页面（带界面接口）

sdk实现调用微博授权页面



实现代码是

```
//如上对socialController进行初始化
UINavigationController *oauthController = [socialController getSocialOAuthC
[self presentViewController:oauthController animated:YES];
```

2.2.5 其他用户操作（不带界面接口）

实现解除授权

```
//如上对socialController进行初始化
[socialController.socialDataService requestUnOAuthWithType:shareToType];
```

实现绑定一个登陆账号，此登陆账号为sdk内用户评论时使用的昵称和头像。如果要绑定的平台一定为已经授权的微博平台，否则无法正常绑定

```
//如上对socialController进行初始化
[socialController.socialDataService requestBindToSnsWithType:shareToType];
```

实现解除登陆账号的绑定

```
//如上对socialController进行初始化
[socialController.socialDataService requestUnOAuthWithType:shareToType];
```

2.3 活动（社会化操作）

活动就是用户的操作，操作方法目前有4类： 分享、评论、 喜欢、浏览。下面逐步介绍各项功能。

2.4 分享

2.4.1 带界面的接口

分享列表页面

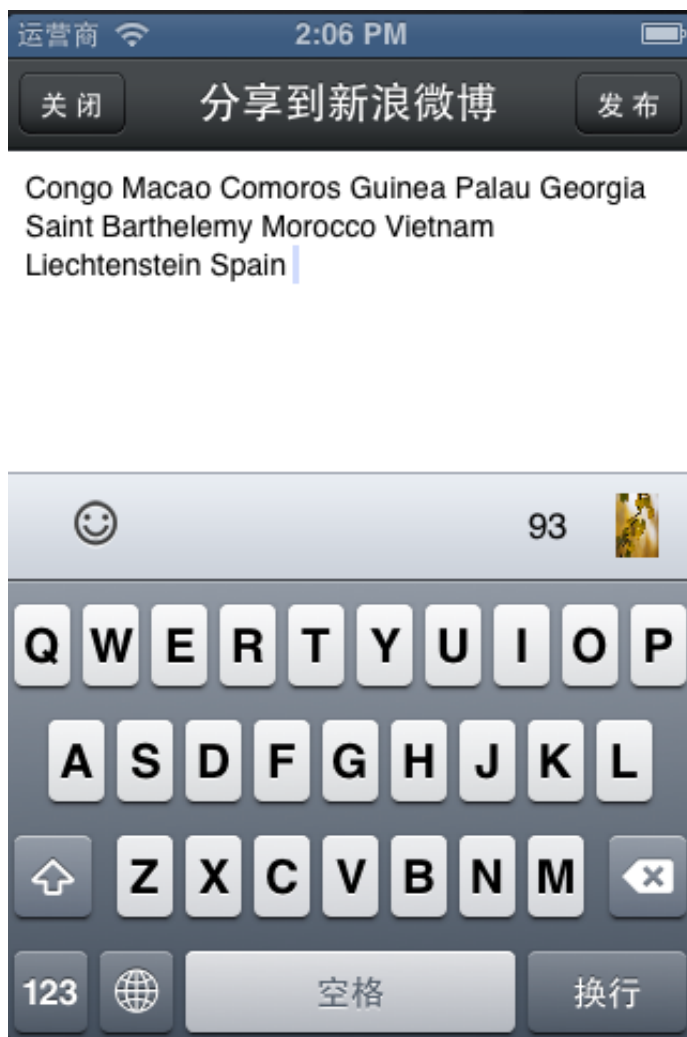


实现代码是

```
UMSocialData *socialData = [[UMSocialData alloc] initWithIdentifier:@"test"]
UMSocialControllerService *socialController = [[UMSocialControllerService c
UINavigationController *shareListController = [socialController getSocialSh
[self presentViewController:shareListController animated:YES];
```

注意，如上面截图所示表格的最后一个单元格，的微信分享不是SDK提供的分享功能，不过利用SDK提供了一个 `<UMSocialUIDelegate>` 的协议，并用此方法 `-(void)setUMSocialUIDelegate:(id <UMSocialUIDelegate>)socialUIDelegate;` 设置实现了此协议的对象。其中实现 `-(UITableViewCell *)customCellForShareListTableView;` 方法可以自定义如图所示的效果，实现 `-(void)didSelectShareListTableViewCell;` 方法是点击此单元格的操作。如果要想实现微信的分享，可以参见我们的demo。

分享微博编辑页面



```
//如上对socialController进行初始化
UINavigationController *shareEditController = [socialController getSocialSh
[self presentViewController:shareEditController animated:YES];
```

2.4.2 不带界面的接口

直接发送微博到对应的微博平台

```
//如上对socialController进行初始化
[socialController.socialDataService postSNSWithType:shareToType usid:nil co
```

其中 `usid` 为分享用的相应的微博平台的id号，如果不传的话而此微博平台使用了 `sdk` 来进行授权过的话，会自动用授权后的 `usid` 进行发送，如果传过来会优先用此 `usid` 号。 `image` 为分享的图片， `location` 为分享地理位置对象。这三个参数可以为空。

同获取微博账号数据一样，如果要得到发送微博成功与否的状态，要实

现 `<UMSocialDataDelegate>` 的 `-(void)didFinishGetUMSocialDataResponse:`

`(UMSocialResponseEntity *)response` 方法。

获取分享数

```
//如上对socialController进行初始化  
[socialController.socialDataService.socialData requestSocialData];
```

同样在回调方法中可以 `response` 对象的类型是 `NSDictionary` 的 `data` 属性中，其中键为 `sn` 的代表分享数。

你也可以获取缓存在本地的分享数

```
[socialController.socialDataService.socialData getShareNumber:UMSNumberShar
```

2.4.3 分享常见错误

在回调方法 `-(void)didFinishGetUMSocialDataResponse:(UMSocialResponseEntity *)response` 中的 `response` 中的 `responseCode` 和 `message` 可以得到返回状态码，和错误消息。

注意如果 `responseCode` 为 `5027` 的话，`message` 是 'Sina access token expired, please do oauth(rebind account) again', 表示授权过期，需要重新打开授权页面进行授权。

2.5 评论

2.5.1 带界面的接口

评论列表



```
//如上对socialController进行初始化
UINavigationController *commentList = [socialController getSocialCommentList
[self presentViewController:commentList animated:YES];
```

2.5.2 不带界面的接口

获取最近的评论数据

```
//如上对socialController进行初始化
[socialController.socialDataService requestCommentList:(-1)];
```

如果评论数据比较多，想获取之前的评论，可以在 `-(void)requestCommentList:(long long)lastCommentTime;` 中传入指定时间戳之前的评论，具体每条评论的时间戳可以在回调方法 `-(void)didFinishGetUMSocialDataResponse:(UMSocialResponseEntity *)response` 中的 `response` 中得到。

添加一条文本评论

```
//如上对socialController进行初始化  
[socialController.socialDataService postCommentWithContent:@"comment content"]
```

添加一条评论，并添加地理位置信息，或者分享到微博平台，可以带有图片

```
//如上对socialController进行初始化  
NSDictionary *shareToSNS = [NSDictionary dictionaryWithObjectsAndKeys:@"weibo",  
[socialController.socialDataService postCommentWithContent:@"comment content"]];
```

获取评论数

```
//如上对socialController进行初始化  
[socialController.socialDataService.socialData requestSocialData];
```

同样在回调方法中可以 `response` 对象的类型是 `NSDictionary` 的 `data` 属性中，其中键为 `cm` 的代表分享数。

你也可以获取缓存在本地的评论数

```
[socialController.socialDataService.socialData getShareNumber:UMSNumberComment];
```

2.6 喜欢

2.6.1.不带界面的接口

发送一个喜欢的请求，或者如果已经喜欢过的话，发送取消喜欢的请求

```
//如上对socialController进行初始化  
[socialController.socialDataService postAddLikeOrCancel];
```

获取喜欢数

```
//如上对socialController进行初始化  
[socialController.socialDataService.socialData requestSocialData];
```

同样在回调方法中可以 `response` 对象的类型是 `NSDictionary` 的 `data` 属性中，其中键为 `lk` 的代表分享数。

你也可以获取缓存在本地的评论数

```
[socialController.socialDataService.socialData getShareNumber:UMSNumberLike];
```

3 错误代码含义

错误码	错误原因
500	给定uid的用户并不存在
501	绑定账户失败
502	解绑定失败
503	评论失败
504	获取评论失败
505	用户被加入黑名单
506	获取好友失败
507	获取授权url失败
508	解除授权失败
510	分享失败
511	获取bar失败
512	从平台获取用户信息失败
513	url跳转失败
514	从social获取用户信息失败
515	请求的参数错误
516	请求喜欢失败
517	版本号错误

分享错误代码

错误码	错误原因
5001	错误的友盟appkey，或者此appkey没有绑定任何平台
5002	这个友盟appkey已经被禁止
5003	请求的参数中没有uid
5004	未知的错误，具体的错误信息会在log的输出中
5005	访问频率超限，可一会儿再试
5006	请求参数中没有content字段
5007	请求参数中content字段的内容为空
5008	没有上传图片
5009	此友盟的appkey没有绑定对应平台的appkey和appsecret
5010	userid无效，这个用户并没有进行授权

5013	请求的参数中没有appkey这个字段
5014	没有对此用户进行授权
5024	获取access token失败，具体看log输出
5025	获取request token失败，具体看log输出
5027	授权已经过期