

# 友盟社区化组件 v6.1.1

## sdk 介绍与集成

1. 简述：友盟社区化组件提供第三方授权登录和第三方分享功能。

2. 手动集成：

(1) 下载 U-Share SDK，目录如下：

- 1. Document - U-Share SDK文档
- 2. UMSocialDemo - U-Share SDK Demo (如点选下载)
- 3. UMSocial - U-Share SDK核心目录
  - UMSocialSDK - U-Share SDK核心framework
  - UMSocialUI - U-Share 分享UI资源、分享面板framework
  - SocialLibraries - 所选择下载的第三方平台SDK及U-Share链接库

(2) 将 U-Share SDK 的核心目录添加到工程

(3) 添加项目配置：Other Linker Flags 添加-ObjC

(4) 加入依赖系统库 libsqlite3.tbd

根据集成的不同平台加入相关的依赖库，未列出平台则不用添加 添加方式：选中项目Target -> Linked Frameworks and Libraries列表中添加

平台	依赖系统库
新浪微博	SystemConfiguration.framework CoreTelephony.framework ImageIO.framework libsqlite3.tbd libz.tbd
微信	SystemConfiguration.framework CoreTelephony.framework libsqlite3.tbd libc++.tbd libz.tbd
QQ	SystemConfiguration.framework libc++.tbd
Twitter	CoreData.framework

注：Twitter平台加入后需添加TwitterKit.framework/Resources/TwitterKitResources.bundle。

3. 通过 Cocoapods 集成

工程 target 名为 UMSocialDemo，则示例如下：

## 1.2.2 微信、微博及QQ集成示例

若需集成微信、微博和QQ平台，可用以下Podfile配置

```
target 'UMSocialDemo' do
  # U-Share SDK UI模块 (分享面板, 建议添加)
  pod 'UMengUShare/UI'

  # 集成新浪微博
  pod 'UMengUShare/Social/Sina'

  # 集成微信
  pod 'UMengUShare/Social/WeChat'

  # 集成QQ
  pod 'UMengUShare/Social/QQ'
end
```

## 1.2.3 其他

若需要添加所有平台，只需添加以下代码：

```
target 'UMSocialDemo' do
  pod 'UMengUShare/Social'
end
```

添加所有平台会增加应用体积，不建议使用，可用于测试

## 1.2.4 从6.0版更新到6.1报错提示

如有编译错误，如

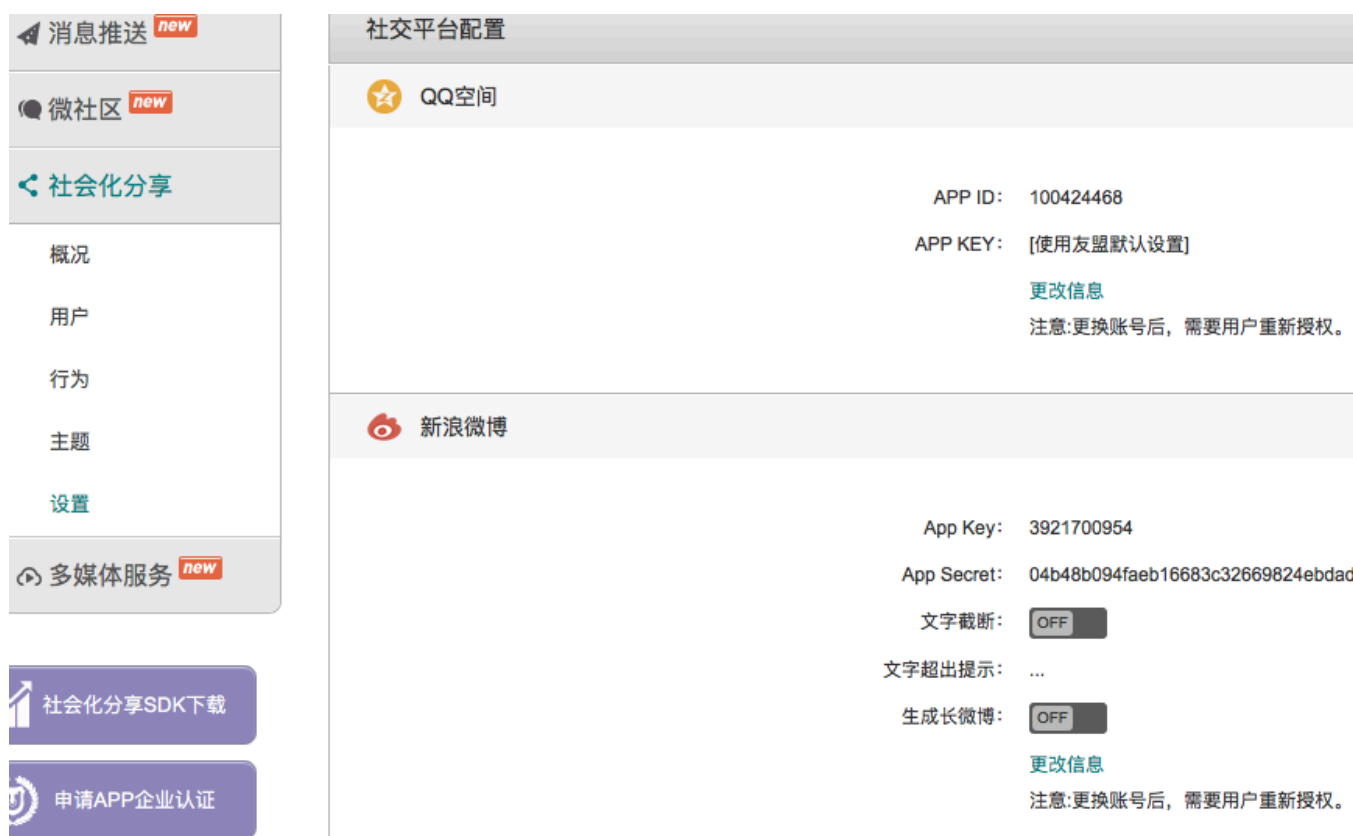
```
Incompatible block pointer types sending 'int (^)(void)'' to parameter of type 'UMSocialSharePlatformSelectionBlock' (aka 'void (^)(UMSocialPlatformType, NSDictionary *__strong)')
```

则说明回调block类型有改动，修改成以下代码即可

```
[UMSocialUIManager showShareMenuViewInWindowWithPlatformSelectionBlock:^(UMSocialPlatformType platformType, NSDictionary *userInfo) {
    // code
}];
```

## 4. U-Share SDK 平台配置

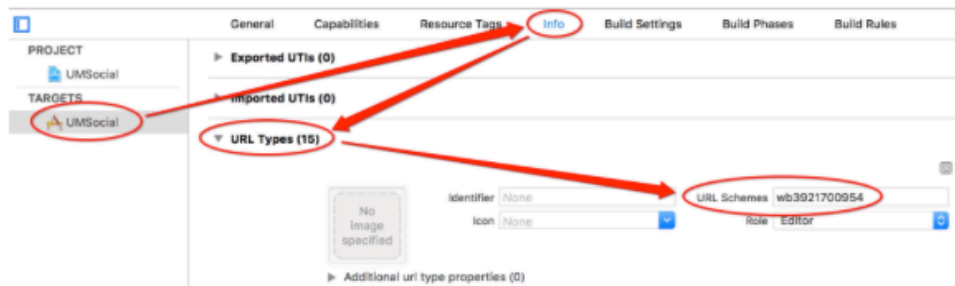
- (1) 需要第三方 appKey 和 appSecret 等信息，可参考第三方账号申请及绑定申请所需的平台账号，绑定第三方帐号信息到友盟后台



### (2) 添加 URL Types

- 1) URL Scheme 是通过系统找到并跳转对应 app 的一类设置，通过向项目中的 info.plist 文件中加入 URL types 可使用第三方平台所注册的 appkey 信息向系统注册你的 app，当跳转到第三方应用授权或分享后，可直接跳转回你的 app。
- 2) 添加 URL Types 有如下几处，都可进行设置

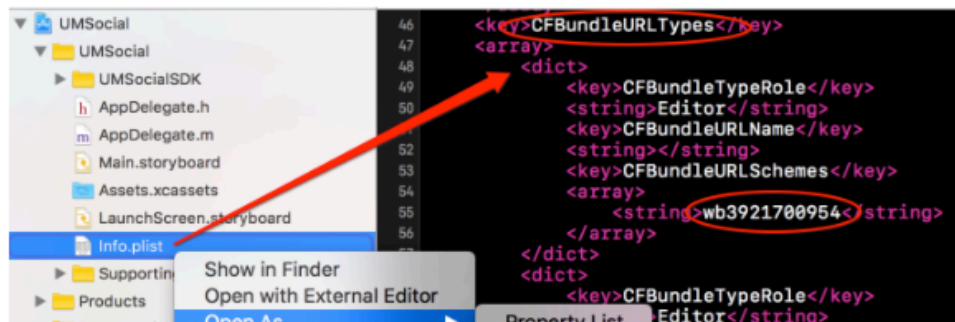
### 1. 通过工程设置面板



### 2. 通过info.plist文件编辑



### 3. 直接编辑info.plist中XML代码



## 3) 配置第三方平台 URL Scheme

未列出则不需设置

平台	格式	举例	备注
微信	微信appKey	wxdc1e388c3822c80b	
QQ/Qzone	以下两项都需添加： 1、"tencent"+腾讯QQ互联应用appID 2、“QQ”+腾讯QQ互联应用appID转换成十六进制（不足8位前面补0）	appID: 100424468 1、tencent100424468 2、QQ05fc5b14	100424468转十六进制为5fc5b14，因不足8位向前补0，结果为05fc5b14，加“QQ”前缀QQ05fc5b14
新浪微博	“wb”+新浪appKey	wb3921700954	
钉钉	钉钉appkey	dingoalmlnohc0wggfedpk	URL Type 中的 identifier填"dingtalk"
支付宝	“ap”+appID	ap2015111700822536	URL Type 中的 identifier 填"alipayShare"
易信	易信appkey	yx35664bdf4db42c2b7be1e29390c1a06	
点点虫	点点虫appID	8112117817424282305	URL Type 中的 identifier填"Laiwang"

### (3) 适配 ios9/10

- 1) 以 iOS10 SDK 编译的工程会默认以 SSL 安全协议进行网络传输，即 HTTPS，如果依然使用 HTTP 协议请求网络会报系统异常并中断请求。目前可用如下两种方式保持用 HTTP 进行网络连接：（详细域名白名单见官方集成文档）

以iOS10 SDK编译的工程会默认以SSL安全协议进行网络传输，即HTTPS，如果依然使用HTTP协议请求网络会报系统异常并中断请求。目前可用如下两种方式保持用HTTP进行网络连接：

在info.plist中加入安全域名白名单(右键info.plist用source code打开)

```
<key>NSAppTransportSecurity</key>
<dict>
    <!-- 配置允许 http的任意网络Begin-->
    <key>NSAllowsArbitraryLoads</key>
    <true/>
    <!-- 配置允许 http的任意网络End-->
    <key>NSExceptionDomains</key>
    <dict>
        <!-- U-Share必须配置的HTTP白名单-->
        <key>log.umsns.com</key>
        <dict>
            <key>NSIncludesSubdomains</key>
            <true/>
            <key>NSTemporaryExceptionAllowsInsecureHTTPLoads</key>
            <true/>
            <key>NSTemporaryExceptionMinimumTLSVersion</key>
            <string>TLSv1.1</string>
        </dict>
    </dict>
</dict>
```

#### 2) 应用跳转白名单

如果你的应用使用了如 SSO 授权登录或跳转分享功能，在 iOS9/10 下就需要增加一个可跳转的白名单，指定对应跳转 App 的 URL Scheme，否则将在第三方平台判断是否跳转时用到的 canOpenURL 时返回 NO，进而只进行 webview 授权或授权/分享失败。同样在 info.plist 增加：

```
<key>LSApplicationQueriesSchemes</key>
<array>
    <!-- 微信 URL Scheme 白名单-->
    <string>wechat</string>
    <string>weixin</string>

    <!-- 新浪微博 URL Scheme 白名单-->
    <string>sinaweibohd</string>
    <string>sinaweibo</string>
    <string>sinaweibosso</string>
    <string>weibosdk</string>
    <string>weibosdk2.5</string>
</array>
```

```

<!-- QQ、Qzone URL Scheme 白名单-->
<string>mqqapi</string>
<string>mqq</string>
<string>mqq0pensdkSSoLogin</string>
<string>mqqconnect</string>
<string>mqqopensdkdataline</string>
<string>mqqopensdkgrouptribeshare</string>
<string>mqqopensdkfriend</string>
<string>mqqopensdkkapi</string>
<string>mqqopensdkkapiV2</string>
<string>mqqopensdkkapiV3</string>
<string>mqqopensdkkapiV4</string>
<string>mqzoneopensdk</string>
<string>wtloginmqq</string>
<string>wtloginmqq2</string>
<string>mqqwpa</string>
<string>mqzone</string>
<string>mqzonev2</string>
<string>mqzoneshare</string>
<string>wtloginqzone</string>
<string>mqzonewx</string>
<string>mqzoneopensdkkapiV2</string>
<string>mqzoneopensdkkapi19</string>
<string>mqzoneopensdkkapi</string>
<string>mqqbrowser</string>
<string>mttbrowser</string>

```

▶ LSAApplicationQueriesSchemes	▲	Array	(32 items)
▶ App Transport Security Settings	▲	Dictionary	(2 items)
▼ URL types	▲	Array	(5 items)
▼ Item 0 (Editor)	▲	Dictionary	(3 items)
URL identifier	▲	String	qq
Document Role	▲	String	Editor
▼ URL Schemes	▲	Array	(1 item)
Item 0	▲	String	tencent100424468
▼ Item 1 (Editor)	▲	Dictionary	(3 items)
URL identifier	▲	String	qzoneScheme
Document Role	▲	String	Editor
▼ URL Schemes	▲	Array	(1 item)
Item 0	▲	String	mqzone
▼ Item 2 (Editor)	▲	Dictionary	(3 items)
URL identifier	▲	String	QQ
Document Role	▲	String	Editor
▼ URL Schemes	▲	Array	(1 item)
Item 0	▲	String	QQ05fc5b14
▼ Item 3 (Editor)	▲	Dictionary	(3 items)
URL identifier	▲	String	weibo
Document Role	▲	String	Editor
▼ URL Schemes	▲	Array	(1 item)
Item 0	▲	String	wb3921700954
▼ Item 4 (Editor)	▲	Dictionary	(3 items)
URL identifier	▲	String	weixin
Document Role	▲	String	Editor
▼ URL Schemes	▲	Array	(1 item)
Item 0	▲	String	wxdc1e388c3822c80b



## 5. 初始化设置

### (1) 初始化 U-Share 及第三方平台

app启动后进行U-Share和第三方平台的初始化工作

```
// 打开调试日志
[[UMSocialManager defaultManager] openLog:YES];

// 设置友盟appkey
[[UMSocialManager defaultManager] setUmSocialAppkey:@"57b432afe0f55a9832001a0a"]
;

// 获取友盟social版本号
//NSLog(@"UMeng social version: %@", [UMSocialGlobal umSocialSDKVersion]);

// 设置微信的appKey和appSecret
[[UMSocialManager defaultManager] setPlaform:UMSocialPlatformType_WechatSession
appKey:@"wxd1e388c3822c80b" appSecret:@"3baf1193c85774b3fd9d18447d76cab0"
redirectURL:@"http://mobile.umeng.com/social"];

// 设置分享到QQ互联的appKey和appSecret
[[UMSocialManager defaultManager] setPlaform:UMSocialPlatformType_QQ appKey:@"1
00424468" appSecret:nil redirectURL:@"http://mobile.umeng.com/social"];

// 设置新浪的appKey和appSecret
[[UMSocialManager defaultManager] setPlaform:UMSocialPlatformType_Sina appKey:@"
3921700954" appSecret:@"04b48b094faeb16683c32669824ebdad" redirectURL:@"http://sns.
whalecloud.com/sina2/callback"];

// 支付宝的appKey
[[UMSocialManager defaultManager] setPlaform:UMSocialPlatformType_AlipaySession
appKey:@"2015111700822536" appSecret:nil redirectURL:@"http://mobile.umeng.com/social"];

```

### (2) 设置系统回调

```
// 支持所有iOS系统
- (BOOL)application:(UIApplication *)application openURL:(NSURL *)url sourceApplication:(NSString *)sourceApplication annotation:(id)annotation
{
    BOOL result = [[UMSocialManager defaultManager] handleOpenURL:url];
    if (!result) {
        // 其他如支付等SDK的回调
    }
    return result;
}

```

注：以上为建议使用的系统 `openURL` 回调，且“新浪”平台仅支持以上回调。还有以下两种回调方式，如果开发者选取以下回调，也请补充相应的函数调用。

1. 仅支持iOS9以上系统，iOS8及以下系统不会回调

```
- (BOOL)application:(UIApplication *)app openURL:(NSURL *)url options:(NSDictionary<UIApplicationOpenURLOptionsKey, id> *)options
{
    BOOL result = [[UMSocialManager defaultManager] handleOpenURL:url];
    if (!result) {
        // 其他如支付等SDK的回调
    }
    return result;
}
```

2.支持目前所有iOS系统

```
- (BOOL)application:(UIApplication *)application handleOpenURL:(NSURL *)url
{
    BOOL result = [[UMSocialManager defaultManager] handleOpenURL:url];
    if (!result) {
        // 其他如支付等SDK的回调
    }
    return result;
}
```

## sdk 具体接口调用

### (1) 第三方平台授权

#### 3.2.1 支持授权的平台

微信、QQ、新浪微博、腾讯微博、人人网、豆瓣、Facebook、Twitter、Linkedin领英、Kakao。

#### 3.2.2 授权并获取用户信息

```
// 在需要进行获取用户信息的UIViewController中加入如下代码
#import <UMSocialCore/UMSocialCore.h>

- (void)getUserInfoForPlatform:(UMSocialPlatformType)platformType
{
    [[UMSocialManager defaultManager] getUserInfoWithPlatform:platformType currentViewController:self completion:^(id result, NSError *error) {
        UMSocialUserInfoResponse *userinfo =result;
        NSString *message = [NSString stringWithFormat:@"name: %@\n icon: %@\n gender: %@\n",userinfo.name,userinfo.iconurl,userinfo.gender];
        UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"UserInfo"
                                                            message:message
                                                            delegate:nil
                                                            cancelButtonTitle:NSLocalizedString(@"确定", nil)
                                                            otherButtonTitles:nil];
        [alert show];
    }];
}
```



## (2) 第三方平台分享

### 1) 弹出分享面板

```
#import "UMSocialUIManager.h"
// 点击分享按钮
- (IBAction)share:(id)sender {
    __weak typeof(self) weakSelf = self;
    // 显示分享面板
    [UMSocialUIManager showShareMenuViewInView:nil
 sharePlatformSelectionBlock:^(UMSocialShareSelectionView *shareSelectionView, NSIndexPath *indexPath, UMSocialPlatformType platformType) {
        // 根据platformType调用相关平台进行分享
    }];
}
```

为避免应用审核被拒，仅会对有效的平台进行显示，如平台应用未安装，或平台应用不支持等会进行隐藏。由于以上原因，在模拟器上部分平台会隐藏。

### 2) 设置分享内容

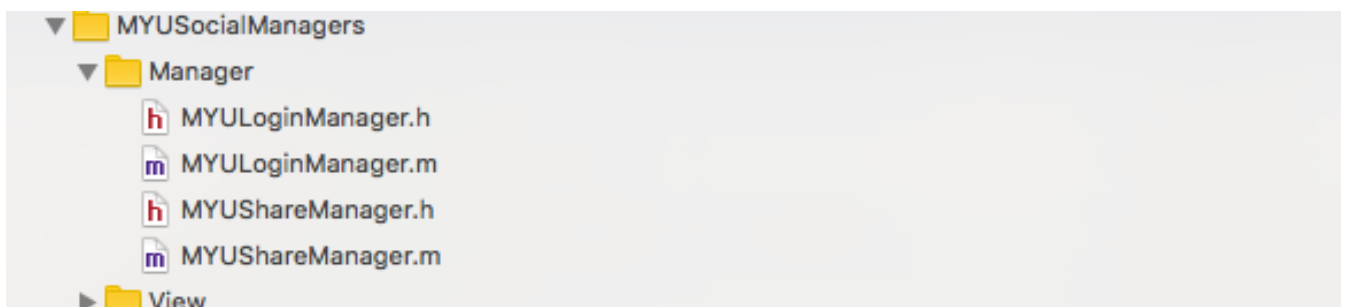
#### 3.3.2.1 分享文本

```
- (void)shareTextToPlatformType:(UMSocialPlatformType)platformType
{
    // 创建分享消息对象
    UMSocialMessageObject *messageObject = [UMSocialMessageObject messageObject];
    // 设置文本
    messageObject.text = @"社会化组件UShare将各大社交平台接入您的应用，快速武装App。";

    // 调用分享接口
    [[UMSocialManager defaultManager] shareToPlatform:platformType messageObject:messageObject
    currentViewController:self completion:^(id data, NSError *error) {
        if (error) {
            NSLog(@"*****Share fail with error %@",error);
        }else{
            NSLog(@"response data is %@",data);
        }
    }];
}
```

其他分享类型示例请参考[U-Share API文档](#)。

## (3) 分享和授权登录封装和使用



## 1) MYULoginManager

```
#import <Foundation/Foundation.h>
#import <UMSocialCore/UMSocialCore.h>

@interface MYULoginManager : NSObject

/**
 * 获取单例对象
 */
+ (id)defaultManager;

/**
 * 授权状态判断
 *
 * @param platform 平台
 */
- (BOOL)isAuth:(UMSocialPlatformType)platformType;

/**
 * 授权操作
 *
 * @note 在isAuth: 返回 NO时进行
 * @param platform 平台
 * @param currentVC 授权操作所在页面, 只对sms,email等平台需要传入viewController的平台, 其他不需要的平台可以传入nil
 * @param completion 授权结果的回调, 含uid, token等信息
 */
- (void)authWithPlatform:(UMSocialPlatformType)platformType currentVC:(UIViewController *)controller completion:(void (^)(UMSocialAuthResponse *result, NSError *error))completion;

/**
 * 取消授权操作
 *
 * @note 在isAuth: 返回 YES时进行
 * @param platform 平台
 * @param completion 取消授权结果的回调
 */
- (void)cancelAuthWithPlatform:(UMSocialPlatformType)platformType completion:(UMSocialRequestCompletionHandler)completion;

/**
 * 获取用户信息
 *
 * @note 在授权后进行获取
 * @param platform 平台
 * @param currentVC 授权操作所在页面
 * @param completion 信息获取结果的回调
 */
- (void)getUserInfoWithPlatform:(UMSocialPlatformType)platformType currentVC:(UIViewController *)currentViewController completion:(void (^)(UMSocialUserInfoResponse *, NSError *error))completion;

#import "MYULoginManager.h"

static MYULoginManager *instance;

@implementation MYULoginManager

+ (instancetype)allocWithZone:(struct _NSZone *)zone {
    static dispatch_once_t onceToken;
    dispatch_once(&onceToken, ^{
        instance = [super allocWithZone:zone];
    });
    return instance;
}

#pragma mark public methods

+ (id)defaultManager {
    static dispatch_once_t onceToken;
    dispatch_once(&onceToken, ^{
        instance = [[MYULoginManager alloc] init];
    });
    return instance;
}

- (BOOL)isAuth:(UMSocialPlatformType)platformType {
    return [[UMSocialDataManager defaultManager] isAuth:platformType];
}
```

```

- (void)authWithPlatform:(UMSocialPlatformType)platformType currentVC:(UIViewController *)controller completion:(void (^)(UMSocialAuthResponse *, NSError *))
completion {
    [[UMSocialManager defaultManager] authWithPlatform:platformType currentViewController:controller completion:^(id result, NSError *error) {
        UMSocialAuthResponse *authresponse = result;
        if(completion){
            completion(authresponse, error);
        }
    }];
}

- (void)cancelAuthWithPlatform:(UMSocialPlatformType)platformType completion:(UMSocialRequestCompletionHandler)completion {
    [[UMSocialManager defaultManager] cancelAuthWithPlatform:platformType completion:^(id result, NSError *error) {
        if(completion){
            completion(result, error);
        }
    }];
}

- (void)getUserInfoWithPlatform:(UMSocialPlatformType)platformType currentVC:(UIViewController *)currentViewController completion:(void (^)(
UMSocialUserInfoResponse *, NSError *))completion {
    [[UMSocialManager defaultManager] getUserInfoWithPlatform:platformType currentViewController:currentViewController completion:^(id result, NSError *error) {
        UMSocialUserInfoResponse *userinfo = result;
        if(completion){
            completion(userinfo, error);
        }
    }];
}

```

## 2) MYUShareManager

```

#import <Foundation/Foundation.h>
#import <UMSocialCore/UMSocialCore.h>

typedef void(^MYUShareCompletionBlock)(id result, NSError *error);

@interface MYUShareManager : NSObject

/**
 * 标题
 * @note 标题的长度依各个平台的要求而定
 */
@property (nonatomic, copy) NSString *title;

/**
 * 描述
 * @note 描述内容的长度依各个平台的要求而定
 */
@property (nonatomic, copy) NSString *descr;

/**
 * 缩略图
 * @note UIImage或者NSData类型或者NSString类型（图片url）
 */
@property (nonatomic, strong) id thumbImage;

/**
 * 获取单例对象
 */
+ (id)defaultManager;

/**
 * 纯文本分享
 *
 * @param text 分享的文本内容
 * @param type 分享的平台
 * @param currentVC 分享所在controller，，只对sms,email等平台需要传入viewController的平台，其他不需要的平台可以传入nil
 * @param completion 分享结果的回调
 */
- (void)shareText:(NSString *)text platformType:(UMSocialPlatformType)type currentVC:(UIViewController *)controller completion:(MYUShareCompletionBlock)
completion;

```

```

/**
 * 图片或图文分享
 * @note 由于iOS系统限制(iOS9+), 非HTTPS的URL图片可能会分享失败, 请使用HTTPS的 URL
 *       图片分享参数可设置URL、NSData类型
 *
 * @param image 分享的图片
 * @param type 分享的平台
 * @param currentVC 分享所在controller
 * @param completion 分享结果的回调
 */
- (void)shareImage:(NSString *)image platformType:(UMSocialPlatformType)type currentVC:(UIViewController *)controller completion:(MYUShareCompletionBlock)
completion;

/**
 * 音乐分享
 *
 * @param musicUrl 分享的音乐地址
 * @param type 分享的平台
 * @param currentVC 分享所在controller
 * @param completion 分享结果的回调
 */
- (void)shareMusicUrl:(NSString *)musicUrl platformType:(UMSocialPlatformType)type currentVC:(UIViewController *)controller completion:(MYUShareCompletionBlock)
completion;

```

```

/**
 * 视频分享
 *
 * @param videoUrl 分享的视频地址
 * @param type 分享的平台
 * @param currentVC 分享所在controller
 * @param completion 分享结果的回调
 */
- (void)shareVideoUrl:(NSString *)videoUrl platformType:(UMSocialPlatformType)type currentVC:(UIViewController *)controller completion:(MYUShareCompletionBlock)
completion;

```

```

static MYUShareManager *instance;

```

```

@implementation MYUShareManager

```

```

+ (instancetype)allocWithZone:(struct _NSZone *)zone {
    static dispatch_once_t onceToken;
    dispatch_once(&onceToken, ^{
        instance = [super allocWithZone:zone];
    });
    return instance;
}

- (instancetype)init {
    if(self = [super init]) {
        _title = @"";
        _descr = @"";
        _thumbImage = nil;
    }
    return self;
}

```

```

#pragma mark public methods

```

```

+ (id)defaultManager {
    static dispatch_once_t onceToken;
    dispatch_once(&onceToken, ^{
        instance = [[MYUShareManager alloc] init];
    });
    return instance;
}

- (void)shareText:(NSString *)text platformType:(UMSocialPlatformType)type currentVC:(UIViewController *)controller completion:(MYUShareCompletionBlock)
completion {
    // 分享数据对象
    UMSocialMessageObject *messageObject = [UMSocialMessageObject messageObject];
    messageObject.text = text;

    [[UMSocialManager defaultManager] shareToPlatform:type messageObject:messageObject currentViewController:controller completion:^(id result, NSError *error) {
        if(completion){
            completion(result, error);
        }
    }];
}

```

```

- (void)shareImage:(NSString *)image platformType:(UMSocialPlatformType)type currentVC:(UIViewController *)controller completion:(MYUShareCompletionBlock)
completion {
    // 分享数据对象
    UMSocialMessageObject *messageObject = [UMSocialMessageObject messageObject];

    UMShareImageObject *shareObject = [UMShareImageObject shareObjectWithTitle:_title descr:_descr thumbImage:_thumbImage];
    [shareObject setShareImage:image];
    messageObject.shareObject = shareObject;

    [[UMSocialManager defaultManager] shareToPlatform:type messageObject:messageObject currentViewController:controller completion:^(id result, NSError *error) {
        if(completion){
            completion(result, error);
        }
    }];
}

```

```

- (void)shareMusicUrl:(NSString *)musicUrl platformType:(UMSocialPlatformType)type currentVC:(UIViewController *)controller completion:(MYUShareCompletionBlock)
completion {
    // 分享数据对象
    UMSocialMessageObject *messageObject = [UMSocialMessageObject messageObject];

    UMShareMusicObject *shareObject = [UMShareMusicObject shareObjectWithTitle:_title descr:_descr thumbImage:_thumbImage];
    [shareObject setMusicUrl:musicUrl];
    messageObject.shareObject = shareObject;

    [[UMSocialManager defaultManager] shareToPlatform:type messageObject:messageObject currentViewController:controller completion:^(id result, NSError *error) {
        if(completion){
            completion(result, error);
        }
    }];
}

```

```

- (void)shareVideoUrl:(NSString *)videoUrl platformType:(UMSocialPlatformType)type currentVC:(UIViewController *)controller completion:(MYUShareCompletionBlock)
completion {
    // 分享数据对象
    UMSocialMessageObject *messageObject = [UMSocialMessageObject messageObject];

    UMShareVideoObject *shareObject = [UMShareVideoObject shareObjectWithTitle:_title descr:_descr thumbImage:_thumbImage];
    [shareObject setVideoUrl:videoUrl];
    messageObject.shareObject = shareObject;

    [[UMSocialManager defaultManager] shareToPlatform:type messageObject:messageObject currentViewController:controller completion:^(id result, NSError *error) {
        if(completion){
            completion(result, error);
        }
    }];
}

```