title: SMShareModel date: 2018-04-03 09:49:20 tags:

- ushow
- share

SMShareModel 简介

该分享模块是一个相对独立的功能模块，主要负责内容分享. 目前可分享的平台如下：

- whatsapp、facebook、Instagram、Messenager、Twitter
- Message(短信)、Email、CopyLink、More、
- StartMarker、Friends

分享类型：

1. 普通分享
2. 强分享
3. h5分享

代码目录结构

```
.
├── SMH5UploadShare(web分享)
│   ├── SMH5GuideShareScollerView
│   ├── SMH5GuideShareViewController
│   ├── SMH5UploadShareScollerView
│   ├── SMH5UploadViewController
│   ├── SMWebGuideTopShareView
│   ├── SMWebShareDataSource
│   └── SMWebShareItemModel
├── SMShare
│   ├── SMCustomShareView
│   ├── SMShareCompetitionActivitiesInfoModel
│   ├── SMShareDynamicConfigurationModel
│   ├── SMShareFriendsModel
│   ├── SMShareFriendsViewController
│   ├── SMShareFriensDataSource
│   ├── SMShareFriensTableViewCell
│   ├── SMShareItemModel
│   ├── SMShareItemSucessState
│   ├── SMShareList.plist
│   ├── SMShareManager
│   ├── SMShareScrollView
│   ├── SMShareVideoDownloadProgressView
│   ├── SMShareVideoDownloadTransitionAnimator
│   ├── SMShareVideoDownloadViewController
│   └── SMShareWebService
├── SMShareAPIUI
```

```
    │    ├── SMShareActivityGuideView
    │    ├── SMShareActivityGuideViewController
    │    ├── SMShareInstrumentLikesView
    │    ├── SMShareItemButton
    │    ├── SMShareOrdinaryContentView
    │    ├── SMShareOrdinaryView
    │    ├── SMShareOrdinaryViewController
    │    ├── SMShareToChannelModel
    │    ├── SMShareViewController
    │    ├── SMStrongShareCell+Rate
    │    ├── SMStrongShareCell+ad
    │    ├── SMStrongShareCell+winIpad
    │    ├── SMStrongShareCell
    │    ├── SMStrongShareHeaderView
    │    ├── SMStrongShareViewController
    │    └── SMStrongShareWinIpadView
    ├── SMShareBounced
    │    ├── SMShareBouncedView
    │    └── SMShareBouncedViewController
    ├── SMShareManager(核心)
    │    ├── SMMFMessageViewController
    │    ├── SMShareAPIContent
    │    ├── SMShareAPIManager
    │    ├── SMShareHelper
    │    │    ├── SMShareAppContentManager
    │    │    ├── SMShareContentHelper
    │    │    └── SMShareHelper
    │    └── SMShareTypeDefine
    ├── SMShareModel
    │    ├── SMShareFromContent
    │    ├── SMShareTrackModel
    │    ├── SMStrongShareDataSource
    │    └── SMStrongShareModel
    ├── SMSharePublishSuccess（发布成功后的分享视图）
    │    ├── SMActivityPublishView
    │    ├── SMSharePublishErrorView
    │    ├── SMSharePublishFailureView
    │    ├── SMSharePublishSuccessView
    │    └── SMSharePublishView
    └── SMStrongShare(强分享)
         ├── Controller
         │    ├── SMStrongChorusViewController
         │    └── SMStrongSoloViewController
         ├── Model
         │    ├── SMShareStrongSoloItemModel
         │    └── SMStrongChorusDataSource
         └── View
              ├── SMStrongChorusHeadView
              ├── SMStrongChorusNaverView
              ├── SMStrongScollerView
              ├── SMStrongSoloExcellentShareView
              └── SMStrongSoloOrdinaryShareView
```

主要使用对象类：

1. 公共分享控件
    - SMShareOrdinaryViewController （目前使用该接口）
    - SMShareViewController（未来将使用这个入口）
    - SMShareFriendsViewController（好友私信分享）
    - SMStrongSoloViewController/SMStrongChorusViewController(独唱/合唱分享入口)
2. 分享管理
    - SMShareAPIManager
3. SMShareTypeDefine 类型定义
   该类中主要定义了一些公共的枚举数据类型：
    1. 分享平台（SMSharePlatformType）
    2. 分享内容类型（SMShareContentType）
    3. 分享状态（SMShareResponseState）
4. 数据模型
    - SMShareAPIContent
    - SMShareItemModel（要分享的数据内容对象）

使用方式

分享控件接口

普通分享

1. SMShareOrdinaryViewController

```
    - (instancetype)initWithShareItme:(SMShareItemModel *)aItemModel
                           returnBack:(ReturnBack)returnBack;

/**
 SMShareModuleTypeVipInvite 这种类型才需要外部赋值
*/
- (instancetype)initWithShareItem:(SMShareItemModel *)aItemModel
                    inviteAPIContent:(SMShareAPIContent *)apiContent
                        returnBack:(ReturnBack)returnBack;
```

2. SMShareViewController

```
 /**
    自定义分享
    目前只支持已知的分享平台类型,之后会增加自定义分享平台的逻辑
    @param shareItems
    分享按钮的item
    例如:
    {
    shareTitle:@"Copy Link",
    shareImage:@"CopylinkImage",
    shareHighlightedImage:@"CopylinkImage",
```

```
    }
    @param shareContent 分享的内容
    @param returnBack
    @return
    */
    - (instancetype)initWithShareItems:(NSMutableArray <NSDictionary
*>*)shareItems
                        shareContent:(SMShareAPIContent *)shareContent
                            returnBack:(ReturnBack)returnBack;
```

强分享

1. SMStrongSoloViewController（独唱分享）

```
@interface SMStrongSoloViewController : SMSNSBaseViewController
//视频是否可以分享的状态
@property (assign, nonatomic) BOOL isShareVideoState;

- (instancetype)initWithShareStrongSoloItemModel:
(SMShareStrongSoloItemModel *)itemModel
                                      returnBlock:
(ReturnBlock)returnBlock;

@end
```

2. SMStrongChorusViewController（合唱分享）

```
    @interface SMStrongChorusViewController : SMSNSBaseViewController

    //视频是否可以分享的状态
    @property (assign, nonatomic) BOOL isShareVideoState;

    - (instancetype)initWithShareStrongSoloItemModel:
(SMShareStrongSoloItemModel *)itemModel
                            returnBlock:(ReturnBlock)returnBlock;

    @end
```

SMShareAPIManager

该分享管理器提供了各平台单独的分享内容接口，在使用过程中调用相应的方法并传入正确的参数即可。

```
    /**
    分享 （目前这个是支持的旧的分享 随时可能废弃 如果要调用分享建议使用下面的方法）

    @required aContent 分享的内容
    @param aContentType 分享内容的类型 （目前只有Instagram 需要区分分享图片或者视频
其他分享平台暂时未用到可以使用SMShareContentTypeAuto）
    @param aPlatformType 分享平台类型
    @required fromViewController 当前控制器
    @param aStateChangedHandler 返回结果
    */
    - (void)initWithShareContent:(SMShareAPIContent *)aContent
                  shareItemModel:(SMShareItemModel *)aShareItemModel
              fromViewController:(UIViewController *)aFromViewController
          popPresenterSourceView:(UIView *)aSourceView
                    platformType:(SMSharePlatformType)aPlatformType
                  onStateChanged:
(SMShareCallApiStateChangedHandler)aStateChangedHandler;


    /**
    分享
    @required aContent 分享的内容
    @required aFromViewController 当前的viewController
    @required aSourceView 当前的来源视图
    @param aPlatformType 分享平台类型
    @param aStateChangedHandler 返回结果
    */
    - (void)initWithShareContent:(SMShareAPIContent *)aContent
              fromViewController:(UIViewController *)aFromViewController
          popPresenterSourceView:(UIView *)aSourceView
                    platformType:(SMSharePlatformType)aPlatformType
                  onStateChanged:
(SMShareCallApiStateChangedHandler)aStateChangedHandler;
    /**
    whatsApp分享

    @required message 分享内容 需要用到shareTitle + shareLink
    */
    - (void)shareWhatsAppWithShareContent:(SMShareAPIContent *)aContent
                           onStateChanged:
(SMShareCallApiStateChangedHandler)aStateChangedHandler;


    /**
    Facebook分享

    @required aContent  shareLink + shareHashtag (link 为必需的 hastag 可选)
    @required fromViewController 当前控制器
    @param aStateChangedHandler
    */
    - (void)shareFacebookWithOpenGraphContent:(SMShareAPIContent
*)aContent
                           fromViewController:(UIViewController
```

```
*)fromViewController
                              onStateChanged:
(SMShareCallApiStateChangedHandler)aStateChangedHandler;


    /**
    Messenger分享

    @required aContent shareLink
    @param aStateChangedHandler
    */
    - (void)shareMessengerWithOpenGraphContent:(SMShareAPIContent
*)aContent
                                  onStateChanged:
(SMShareCallApiStateChangedHandler)aStateChangedHandler;


    /**
    Instagram分享

    @required aContent 图片地址 shareVieoURL 或者 视频
    @param aStateChangedHandler
    */
    - (void)shareInstagramWithContent:(SMShareAPIContent *)aContent
                     onStateChanged:
(SMShareCallApiStateChangedHandler)aStateChangedHandler;


    /**
    分享到 Twitter

    @required aContent shareTitle +  shareLink +  shareImageURL (必须保证至少
一个不是空的)
    @param aStateChangedHandler
    */
    - (void)shareTwitterContent:(SMShareAPIContent *)aContent
                onStateChanged:
(SMShareCallApiStateChangedHandler)aStateChangedHandler;


    /**
    分享到短信

    @required aContent shareTitle + shareLink
    @required fromViewController
    @param aStateChangedHandler
    */
    - (void)shareTextWithContent:(SMShareAPIContent *)aContent
            fromViewController:(UIViewController *)fromViewController
                onStateChanged:
(SMShareCallApiStateChangedHandler)aStateChangedHandler;


    /**
```

```
    分享到邮件

    @required aContent shareTitle + shareLink + shareDescription
    @required fromViewController
    @param state
    */
    - (void)shareEMailWithContent:(SMShareAPIContent *)aContent
            fromViewController:(UIViewController *)fromViewController
                onStateChanged:
(SMShareCallApiStateChangedHandler)aStateChangedHandler;

    // 自动发布facebook
    - (void)automaticShareFacebookWithOpenGraphContent:(SMShareAPIContent
*)aContent
                                           onStateChanged:
(SMShareCallApiStateChangedHandler)aStateChangedHandler;
    // 当前的FB自动发布权限是否可用
    + (BOOL)isFacebookShareHasGranted;

    // 获取当前的FB自动发布权限
    + (void)facebookShareHasGrantedWithViewController:(UIViewController
*)aFromViewController
                                        onStateChanged:
(SMFacebookSyncToState)aStateChangedHandler;

    // 删除facebook上传权限 (PM 要求内部控制删除权限 不走facebook的删除权限接口 但是
已经实现 如果需要可以把注释掉的内容打开)
    + (void)facebookDeletePublishactionOnStateChanged:
(SMFacebookSyncToState)aStateChangedHandler;

    // 判断Twitter是否登陆
    + (void)isTwitterLogin:(SMShareTwitterState)state;
    ```


SMShareAPIContent （分享内容模型）
1. 属性
    ```objc
    @property (copy, nonatomic) NSString *shareRecordingId; //分享的唯一标识
不要重复
    @property (assign, nonatomic) SMShareContentType
shareContentSourceType; //之后会废弃这个字段
    @property (copy,  nonatomic) NSString *shareTitle; //分享标题
    @property (copy,  nonatomic) NSString *shareDescription; // 分享内容
    @property (copy,  nonatomic) NSString *shareType; //分享类型 invite , 打
点使用
    @property (assign, nonatomic) SMShareContentType shareContentType; //分
享内容类型
    @property (copy,  nonatomic) NSString *shareLink; //分享链接
    @property (copy,  nonatomic) NSString *shareImageURL;//分享图片
    @property (copy,  nonatomic) NSString *shareVieoURL;//分享视频下载地址
    @property (copy,  nonatomic) NSString *shareVideoPath;//分享视频本地地址
    @property (copy,  nonatomic) NSString *shareHashtag; //分享Hashtag
    @property (assign, nonatomic) BOOL isShareVideoType; // 是否是分享视频
    @property (assign, nonatomic) NSInteger shareIspublic;//是否是私密状态
```

```
    @property (copy, nonatomic) NSString *isContested;//是否是参赛作品 1，参赛
0，未参赛


    /****************************************
    * 以下为IM分享携带信息
    * 如果可以取到信息请尽量传入避免之后请求
    *
    ****************************************/

    @property (strong,  nonatomic) SMShareFromContent *fromContent; //分享
发送方信息
    @property (copy, nonatomic) NSString *shareSingID;//分享作品歌手id
    @property (copy, nonatomic) NSString *shareSingName;//分享作品歌手名字
    @property (copy, nonatomic) NSString *shareSingCover;//分享作品歌手头像
url
    @property (assign, nonatomic) BOOL isVerified;//分享作品歌手是否是认证歌手
    @property (nonatomic,assign) NSInteger  isRecordingPublic;//是否是我创建
的作品
    ```
```

便捷方法:

```objc
- (instancetype)initWithShareContentTitle:(NSString *) aTitle
                              description:(NSString *) aDescription
                              recordingId:(NSString *) aRecordingId
                                     link:(NSString *) aLink
                                 imageURl:(NSString *) aImageUrl
                                 videoURL:(NSString *) aVideoUrl
                                  hashtag:(NSString *) ahashtag
                              contentType:(SMShareContentType)
aContentType;
```

SMShareHelper

根据title获取分享平台类型 、获取全部分享平台类型 、分享私信好友状态是否可用等功能接口

```
/**
根据返回传入item的title，获取分享平台类型

@param title 分享的title
@return 分享平台类型
*/
+ (SMSharePlatformType)getShareTypeForTitle:(NSString *)title;

/**
根据分享平台类型返回分享平台名字

@param aPlatformType 分享平台类型
@return 平台名字
*/
+ (NSString *)channelWithPlatformType:(SMSharePlatformType)aPlatformType;
```
```

```
 /**
 根据分享平台类型配置link 规则：link + shareType + appName

 @param link 分享的链接
 @param shareType 分享类型
 @return 返回配置后的分享URL
 */
 + (NSString *)shareTitleInfoCombinationLink:(NSString *)link shareType:
(SMSharePlatformType)shareType;

 /**
 获取可分享平台的Items

 @param songId 判断当前分享歌曲是否在在白名单内，如果不在会默认不显示Instagram分享平台 （如果不要做这个判断直接传空）
 @param imageName 是否需要替换不同的More图片 （因为项目中多个地方使用了不同的moreIcon图片，所以特别加这个接口.如果不需要替换可以不传会显示默认icon）
 @return 返回生产的分享平台的Items信息
 */
 + (NSMutableArray *)shareItemArrayIsCanShareSongID:(NSString *)songId
                                  andMoreImageName:(NSString *)imageName
                                      hideInstagram:(BOOL)hideInstagram;


 + (NSMutableArray *)shareItemKtvAndLiveArrayMoreImageName:(NSString
*)imageName;

 /**
 增加点击分享平台数量

 @param aPlatformType 平台
 */
 + (void)writeToFile:(SMSharePlatformType)aPlatformType;

 /**
 根据分享返回结果返回分享结果

 @param responseState 分享返回的结果
 @return 结果的描述
 */
 + (NSString *)resultWithResponseState:
(SMShareResponseState)responseState;


 /**
 分享私信是否可用

 @return
 */
 - (BOOL)isShareThroughChat;
```

```
 /**
 获取分享应用下载link
 starmaker 和 the voice 使用旧的链接不进行修改

 印度和印尼以及以后的app链接 遵守（ https://m.singit.vip/download + apptype） 规则 具体参考：T23035

 @return
 */
 + (NSString *)getShareAppDownloadLink;/**
 根据返回传入item的title，获取分享平台类型

 @param title 分享的title
 @return 分享平台类型
 */
 + (SMSharePlatformType)getShareTypeForTitle:(NSString *)title;

 /**
 根据分享平台类型返回分享平台名字

 @param aPlatformType 分享平台类型
 @return 平台名字
 */
 + (NSString *)channelWithPlatformType:(SMSharePlatformType)aPlatformType;


 /**
 根据分享平台类型配置link 规则：link + shareType + appName

 @param link 分享的链接
 @param shareType 分享类型
 @return 返回配置后的分享URL
 */
 + (NSString *)shareTitleInfoCombinationLink:(NSString *)link shareType:(SMSharePlatformType)shareType;

 /**
 获取可分享平台的Items

 @param songId 判断当前分享歌曲是否在在白名单内，如果不在会默认不显示Instagram分享平台 （如果不要做这个判断直接传空）
 @param imageName 是否需要替换不同的More图片 （因为项目中多个地方使用了不同的moreIcon图片，所以特别加这个接口.如果不需要替换可以不传会显示默认icon）
 @return 返回生产的分享平台的Items信息
 */
 + (NSMutableArray *)shareItemArrayIsCanShareSongID:(NSString *)songId
                                 andMoreImageName:(NSString *)imageName
                                     hideInstagram:(BOOL)hideInstagram;


 + (NSMutableArray *)shareItemKtvAndLiveArrayMoreImageName:(NSString *)imageName;
```

```
 /**
 增加点击分享平台数量

 @param aPlatformType 平台
 */
 + (void)writeToFile:(SMSharePlatformType)aPlatformType;

 /**
 根据分享返回结果返回分享结果

 @param responseState 分享返回的结果
 @return 结果的描述
 */
 + (NSString *)resultWithResponseState:
(SMShareResponseState)responseState;


 /**
 分享私信是否可用

 @return
 */
 - (BOOL)isShareThroughChat;



 /**
 获取分享应用下载link
 starmaker 和 the voice 使用旧的链接不进行修改

 印度和印尼以及以后的app链接 遵守（ https://m.singit.vip/download + apptype） 规
则 具体参考：T23035

 @return
 */
 + (NSString *)getShareAppDownloadLink;
```

目录结构存在的问题(**已修复**)：

1. SMSharePublisthSuccess 代码存放目录不对
2. SMStrongShare 存放目录不对
3. SMShare 代码存放目录不对

存在的问题

1. https://phabricator.ushow.media/T22065
2. 单个文件目录下的文件太多，按不同类型的功能类分成不同的子文件夹
3. SMSNSManager 与SMShareAPIManager在功能上有相似的地方 ,该模块可以去除SMSNSManager相关方法的使用，改用SMShareAPIManager
4. 独/合唱 使用的是旧接口，可以考虑使用新的分享接口替换
5. 存在循环引用问题，dealloc方法一直没有走

疑问： 该模块引入的SNS模块和SSMLoginManager，主要意图是啥?

Q： 我们项目中很多视图控制器都重写了init方法，如果我们的视图控制器由xib/sb加载，则这个方法是不会走的。

补充:

1. 在项目中隐藏系统的导航栏

```
- (void)viewWillAppear:(BOOL)animated
{
    [super viewWillAppear:animated];
    [self.navigationController setNavigationBarHidden:YES
animated:NO];
}
```

这样在push页面过程中导航栏会出黑色 建议把这个动画打开

```
- (void)viewWillAppear:(BOOL)animated
{
    [super viewWillAppear:animated];
    [self.navigationController setNavigationBarHidden:YES
animated:YES];
}
```

改进的地方

1. 项目中代码的逻辑目录需要有对应的物理文件夹
2. 将代码按不同的功能模块归纳到相应的目录下
3. 重复使用的代码可以考虑提取成一个方法，供使用
4. 一个代码文件里的代码可以按不同的方法类型 使用 `#pragma mark -`进行归纳区分，方便阅读
5. 项目中的注释有点少，.h文件里的接口加上相应的注释，有利于自己和他们阅读与理解

参考文档： ShareSDK