

iOS-----IMSDK文档

项目依赖第三方框架

- XMPPFramework
- RITLPhotos
- MJExtension
- AFNetworking

主要文件说明

- IMBaseSettings.h为IM配置文件,用来配置IM基本信息,如域名,端口号等
- BXClient文件为IM的重要文件,包括初始化,登录,登出,收发消息,好友管理,群组管理等

调用文档说明

1.初始化函数

- options SDK配置项,详细参数查看BXOptions文件

```
- (void)initializeSDKWithOptions: (BXOptions *)options;
```

2.登录/登出IM服务器

在切换到后台或关闭APP时要调用登出,在切换到前台或启动APP时须调用登录

- userId:用户的id
- passWord: 用户的密码(MD5加密)
- completed 回调(success为YES代表成功登录IM服务器,登录失败返回error错误信息)

```
// 登录
- (void)loginWithUserID: (NSString *)userId
    passWord: (NSString *)passWord
    completed: (void(^)(BOOL success, NSString *error))completed;
```

```
// 登出
[[BXClient sharedClient] loginOut];
```

3. 发送消息,单聊,群聊(包括文本信息,表情信息,图片信息,语音信息,视频)

发送语音、图片、文件、视频时,须先将文件上传到服务器,在进行发送

- msg参数类型为WH_JXMessageObject,将参数封装成WH_JXMessageObject发送
 - timeSend: 发送时间
 - fromUserId: 发送方id
 - toUserId: 接收方id
 - isGroup: 是否为群聊
 - content: 发送内容或者URL
 - type: 发送的类型
 - isSend: 是否送达
 - isRead: 是否读取
 - 其他字段详情见WH_JXMessageObject文件
- name: 群聊id

```
- (void)sendMessage:(BXMessage *)msg
    roomName:(NSString * _Nullable)roomName
    completion:(void (^)(NSError * error))completion;
```

4. 接收消息,单聊,群聊

- 注册消息代理

```
[BXClient sharedClient].delegate = self
```

- 接收消息代理回调

```
- (void)messageDidReceive: (XMPPMessage *)message {
    // 可以使用EPMessageTool文件对数据进行解析和拉取数据
    // 接收到的字段
    NSString *type = [[message attributeForName:@"type"] stringValue];
    NSString *messageId = [[message attributeForName:@"id"] stringValue];
    NSString *body = [[message elementForName:@"body"] stringValue];
    NSString *fromUserId=[message fromStr];
    NSString *toUserId=[message toStr];
}
```

5. 上传DeviceToken

```
// 将deviceToken上传到服务器
- (void)bindDeviceToken:(NSData *)token;
```

好友管理,监听状态添加代理

BXContactManagerDelegate,详情见BXContact文件

获取好友列表

```
- (void)getContactsFromServerWithCompletion:(void (^)(NSArray<BXMemberData *> *list,
NSString *errorMsg))completion;
```

加入黑名单

```
- (void)addUserToBlackList:(NSString *)userId
    completion:(void (^)(NSString *userId, NSString *errorMsg))completion;
```

移除黑名单

```
- (void)removeUserToBlackList:(NSString *)userId
    completion:(void (^)(NSString *userId, NSString *errorMsg))completion;
```

添加好友

```
- (void)addContact:(NSString *)userId
    message:(NSString *)message
    completion:(void (^)(NSString *userId, NSString *errorMsg))completion;
```

监听加好友请求

```
/*!
 * 用户A发送加用户B为好友的申请, 用户B会收到这个回调
 *
 * @param userId 用户id
 * @param message 附属信息
 * @param isNeedVeri 是否需要验证添加好友
 */
- (void)friendRequestDidReceiveFromUser:(NSString *)userId
    message:(NSString *)message
    isNeedVeri:(BOOL)isNeedVeri;
```

同意添加好友

```
- (void)approveFriendRequestFromUser:(NSString *)userId
    completion:(void (^)(NSString *userId, NSString
                        *errorMsg))completion;
```

好友监听

```
/// 用户A发送加用户B为好友的申请，用户B同意后，用户A会收到这个回调
/// @param userId 好友id
- (void)friendRequestDidApproveByUser:(NSString *)userId;
```

删除好友

```
- (void)deleteContact:(NSString *)userId
    completion:(void (^)(NSString *userId, NSString *errorMsg))completion;
```

群组管理,监听状态添加代理BXGroupDelegate,详情见BXGroup文件

创建群组

```
- (void)createGroupWithName:(NSString *)groupName
    invitees:(NSArray *)invitees
    message:(NSString *)message
    options:(BXGroupOptions *)options
    completion:(void (^)(BXGroup *group, NSString *errorMsg))completion;
```

成员加入群组

```
- (void)joinRoom:(NSString *)groupId;
```

主动离开群组(群主不能退出群，只能销毁群)

```
- (void)leaveGroup:(NSString *)groupId
    completion:(void (^)(NSString *errorMsg))completion;
```

踢出群组成员

```
- (void)kickGroup:(NSString *)groupId
    userId:(NSString *)userId
  completion:(void (^)(NSString *errorMsg))completion;
```

解散群组

```
- (void)destroyGroup:(NSString *)groupId
  completion:(void (^)(NSString *errorMsg))completion;
```

获取群组列表

```
- (void)getListRoomWithCompletion:(void (^)(
  NSMutableArray<BXGroup *> *listRoom, NSString *errorMsg))completion;
```

获取群成员列表

```
- (void)getGroupMemberListWithGroupId:(NSString *)groupId
                                completion:(void (^)(NSMutableArray<BXMemberData *>
  *listRoom, NSString *errorMsg))completion;
```

邀请用户加入群组

```
- (void)addMembers:(NSArray *)users
    toGroup:(NSString *)groupId
  completion:(void (^)(BXGroup *aGroup, NSString *errorMsg))completion;
```

添加群管理员

```
- (void)addAdmin:(NSString *)admin
    toGroup:(NSString *)groupId
  completion:(void (^)(BXGroup *aGroup, NSString *errorMsg))completion;
```

禁言群组成员

```
- (void)muteMember:(NSString *)muteMember
  muteMilliseconds:(NSInteger)muteMilliseconds
    fromGroup:(NSString *)groupId
  completion:(void (^)(BXGroup *aGroup, NSString *errorMsg))completion;
```

禁言全部群组成员

```
- (void)muteGroup:(NSString *)muteGroup
muteMilliseconds:(NSInteger)muteMilliseconds
completion:(void (^)(BXGroup *aGroup, NSString *errorMsg))completion;
```

群消息免打扰/取消群消息免打扰

```
- (void)shieldGroup:(NSString *)groupId
offlineNoPushMsg:(BOOL)offlineNoPushMsg
completion:(void (^)(BXGroup *aGroup, NSString *errorMsg))completion;
```

获取群信息

```
- (void)getGroupInfoWithId:(NSString *)groupId
completion:(void (^)(BXGroup *aGroup, NSString *errorMsg))completion;
```

更改群名称

```
- (void)updateGroupName:(NSString *)name
forGroup:(NSString *)groupId
completion:(void (^)(BXGroup *aGroup, NSString *errorMsg))completion;
```

更改群描述

```
- (void)updateGroupDesc:(NSString *)desc
forGroup:(NSString *)groupId
completion:(void (^)(BXGroup *aGroup, NSString *errorMsg))completion;
```

更改群主(只有Owner有权限)

```
- (void)updateGroupOwner:(NSString *)groupId
newOwner:(NSString *)newOwner
completion:(void (^)(BXGroup *aGroup, NSString *errorMsg))completion;
```