

# 研修宝“聊聊”集成说明 v1.0

## 拷贝SDK

将IM目录下所有内容拷贝到项目中

## 初始化

用户登录后才能初始化，因为需要用户的信息

// 配置当前用户和token

```
[[IMManager sharedInstance]setupWithCurrentMember: token:];
```

// 配置场景id，一般为当前班级id

```
[[IMManager sharedInstance]setupWithSceneID:];
```

## 启动SDK

用户登录状态时启动，一般放在初始化之后

```
[[IMManager sharedInstance] startConnection];
```

## 停止SDK

用户退出登录或者进入后台时调用，当由后台转前台时，登录用户需要重新启动被停止的SDK

```
[[IMManager sharedInstance] stopConnection];
```

## UI对接

IM对UI层提供的接口都位于IMUserInterface类中，请不要直接使用其它类的方法。

下面三个用于发送文字消息

```
+ (void)sendTextMessageWithText:(NSString *)text topicID:(int64_t)topicID;//
```

新消息

```
+ (void)sendTextMessageWithText:(NSString *)text topicID:(int64_t)topicID
```

```
uniqueID:(NSString *)uniqueID;// 失败消息重发
```

```
+ (void)sendTextMessageWithText:(NSString *)text toMember:(IMMember
```

\*)member fromGroup:(int64\_t)groupId;// 新建会话消息发送, 此时没有topicID

发送图片消息, 逻辑与文字消息类似

```
+ (void)sendImageMessageWithImage:(UIImage *)image topicID:
(int64_t)topicID;
+ (void)sendImageMessageWithImage:(UIImage *)image topicID:
(int64_t)topicID uniqueID:(NSString *)uniqueID;
+ (void)sendImageMessageWithImage:(UIImage *)image toMember:
(IMMember *)member fromGroup:(int64_t)groupId;
```

获取本地缓存的所有话题

```
+ (NSArray<IMTopic *> *)findAllTopics;
```

按序取一定数量的缓存消息, 用于聊天页面第一次展示时

```
+ (void)findMessagesInTopic:(int64_t)topicID
        count:(NSUInteger)count
        asending:(BOOL)asending
        completeBlock:(void(^)(NSArray<IMTopicMessage *> *array, BOOL
hasMore))completeBlock;
```

获取话题的历史消息

```
+ (void)findMessagesInTopic:(IMTopic *)topic
        count:(NSUInteger)count
        beforeMsg:(IMTopicMessage *)msg;
```

清楚话题标记的未读消息数量, 用于聊天页面退出时

```
+ (void)resetUnreadMessageCountWithTopicID:(int64_t)topicID;
```

查找是否有与该人的对话, 用于从通讯录打开与某人的对话

```
+ (IMTopic *)findTopicWithMember:(IMMember *)member;
```

记录server时间和本地时间偏差, 在消息发送中用本地时间加上偏差的值模拟server时间时使用。

```
+ (NSTimeInterval)obtainTimeoffset;
```

判断用户是否在此话题中，用于聊天时尚未建立话题的情况，如果返回YES，则该话题就是与此用户的对话

+ (BOOL)topic:(IMTopic \*)topic isWithMember:(IMMember \*)member;

判断两个topic是否相等，如果相等，需要用新的topic替换掉本地的topic

+ (BOOL)isSameTopicWithOneTopic:(IMTopic \*)topic anotherTopic:(IMTopic \*)anotherTopic;

更新话题基本信息，仅用于更新名称，成员信息

+ (void)updateTopicInfoWithTopicID:(int64\_t)topicID;

**IM对内容的更新统一使用通知的形式，UI层需要监听相应的通知来获得内容的变化以便更新UI。**

kIMMessageDidUpdateNotification

消息更新通知，时机：新消息，消息状态的变化

kIMTopicDidUpdateNotification

话题更新通知，时机：新话题，话题加减成员

kIMTopicInfoUpdateNotification

话题信息更新通知，时机：话题的名称，成员信息的改变

kIMTopicDidRemoveNotification

话题移除通知，时机：话题被移除

kIMUnreadMessageCountDidUpdateNotification

未读消息数量变化通知，时机：话题的未读消息数量变化

kIMImageUploadDidUpdateNotification

图片上传进度更新通知，时机：图片上传进度变化时

kIMHistoryMessageDidUpdateNotification

历史消息获取通知，时机：获取到历史消息时

## 依赖三方库

MQTTClient

Protobuf

MagicalRecord

## 依赖本地封装模块

### **HttpRequest**

本地封装的基于ASIHttpRequest的网络请求模块

### **七牛**

本地封装的基于Qiniu的云端存储模块