

Android SDK介绍

1.配置服务器

UploadFileUrl: 上传文件的服务器地址

代码

```
ConfigBean configBean = new ConfigBean();
    configBean.setXmppDomain(domain);
    configBean.setXmppHost(host);
    configBean.setXmppPort(port);
    configBean.setUploadFileUrl(uploadFileUrl);
IMManager.getInstance().init(this, configBean);
```

2.登录成功后自动连接 IM 服务器

代码

```
IMManager.getInstance().login(this, loginInfo, phone, nickName, password);
```

3.回调

登录状态回调: setOnConnectListener

消息回调: setChatReceiveListener

群操作回调: setMucListener

代码

```
IMManager.getInstance().setOnConnectListener(new OnConnectListener() {
    @Override
    public void onConnect(ConnectSate connectSate, String msg) {
    }
});

IMManager.getInstance().setChatReceiveListener(new ChatMessageListener() {
    @Override
    public void onMessageSendStateChange(ChatMessageNotify messageNotify) {
        //消息状态回调
    }
})
```

```

        @Override
        public void onNewMessage(List<ChatMessage> chatMessages) {
            //收到新消息
        }
    });

    IMManager.getInstance().setMucListener(new MucListener() {
        @Override
        public void onMyBeInvite(String roomName) {
            //我被邀请进群
        }

        @Override
        public void onNewMemberBeInvite(String roomName, String userName) {
            //其他人被邀请进群
        }

        @Override
        public void onMyBeDelete(String roomName) {
            //我被踢出群
        }

        @Override
        public void onOtherBeDelete(String roomName, String toUserId, String
toUserName) {
            //其他人被踢出群
        }

        @Override
        public void onExitMucRoom(String roomName, String toUserId, String
toUserName) {
            //自动退群
        }

        @Override
        public void onRoomNameChange(String roomName) {
            //群名被修改
        }

        @Override
        public void onRoomNoticeChange(String roomId, String roomNotice, String
noticeTime) {
            //发布群公告
        }

        @Override
        public void onSetManager(String toUserId, String userName) {
            //设置管理员

```

```

    }

    @Override
    public void onDeleteMucRoom(String roomName) {
        //删除群
    }

    @Override
    public void onGroupTransfer(String roomName, String toUserId, String
userName) {
        //群转让
    }
});

```

4.退出服务器

代码

```
IMManager.getInstance().logout();
```

5.发送消息 单聊和群聊(包括文本信息、图片、语音、视频、文件、位置、名片)

发送语音、图片、文件、视频等文件消息时，将文件上传到服务器，在进行文件上传

IMManager.getInstance().sendMessage(); 发送单聊

IMManager.getInstance().sendRoomMessage(); 发送群聊

代码

```

ChatMessage chatMessage = ChatMessage.createTxtMessage(toUserID, content);
ChatMessage chatMessage = ChatMessage.createImageMessage(toUserID, imageFilePath);
ChatMessage chatMessage = ChatMessage.createVoiceMessage(toUserID, voiceFilePath,
voiceTimeLen);
ChatMessage chatMessage = ChatMessage.createVideoMessage(toUserID, videoPath);
ChatMessage chatMessage = ChatMessage.createFileMessage(toUserID, filePath);
ChatMessage chatMessage = ChatMessage.createLocationMessage(toUserID,
locationImageFilePath, latitude, longitude, address);
ChatMessage chatMessage = ChatMessage.createCardMessage(toUserID, cardUserName,
cardUserId);

IMManager.getInstance().sendMessage(this, chatMessage);
IMManager.getInstance().sendRoomMessage(this, chatMessage);

```

6.接收消息的监听

onChatReceiveMessageNotify: 发送消息回执回调

onChatReceiveMessage: 接收消息回调

代码

```
IMManager.getInstance().setOnChatReceiveListener(new OnChatReceiveListener() {  
    @Override  
    public void onChatReceiveMessageNotify(ChatMessageNotify messageNotify) {  
    }  
  
    @Override  
    public void onChatReceiveMessage(List<ChatMessage> chatMessages) {  
    }  
});
```

7.创建群

friendList: List friendList 邀请进群的好友集合

new ResponseCallBack: 请求成功失败的回调

代码

```
IMManager.getInstance().createRoom(this, roomName, roomDesc, friendList, new  
ResponseCallBack<String>() {  
    @Override  
    public void onSuccess(String result) {  
    }  
  
    @Override  
    public void onFailure(String result) {  
    }  
});
```

8.解散群

代码

```
IMManager.getInstance().roomDelete(roomId, new ResponseCallBack<String>() {  
    @Override  
    public void onSuccess(String result) {  
    }  
  
    @Override  
    public void onFailure(String result) {  
    }  
});
```

9.群加好友

friendList: List friendList 邀请进群的好友集合

代码

```
IMManager.getInstance().inviteRoomFriend(roomId, friendList, new
ResponseCallBack<String>() {
    @Override
    public void onSuccess(String result) {
    }

    @Override
    public void onFailure(String result) {
    }
});
```

10.群踢人或自己退群

userId: 退群用户的 id

代码

```
IMManager.getInstance().roomMemberDelete(roomId, userId, new ResponseCallBack<String>
() {
    @Override
    public void onSuccess(String result) {
    }

    @Override
    public void onFailure(String result) {
    }
});
```

11.群主转让

toUserId: 转让给某用户的 id

代码

```

IMManager.getInstance().roomTransfer(roomId, toUserId, new ResponseCallBack<String>()
{
    @Override
    public void onSuccess(String result) {
    }

    @Override
    public void onFailure(String result) {
    }
});

```

12. 获取群信息

代码

```

IMManager.getInstance().getRoomMemberForAvatar(roomId, new ResponseCallBack<MucRoom>()
{
    @Override
    public void onSuccess(MucRoom mucRoom) {
    }

    @Override
    public void onFailure(String result) {
    }
});

```

13. 获取群列表

代码

```

IMManager.getInstance().getRoomList(new ResponseCallBack<List<MucRoom>>() {
    @Override
    public void onSuccess(List<MucRoom> mucRooms) {
    }

    @Override
    public void onFailure(String result) {
    }
});

```

14. 获取成员列表

代码

```
IMManager.getInstance().getRoomMemberList(roomId, new
ResponseCallBack<List<MucRoomMember>>() {
    @Override
    public void onSuccess(List<MucRoomMember> mucRoomMembers) {
    }

    @Override
    public void onFailure(String result) {
    }
});
```

15.获取自己的成员信息以及群属性

代码

```
IMManager.getInstance().getRoomMyDetail(roomId, new ResponseCallBack<MucRoom>() {
    @Override
    public void onSuccess(MucRoom mucRoom) {
    }

    @Override
    public void onFailure(String result) {
    }
});
```

16.修改群名称或群公告

修改群名称传roomName参数，roomNotice传""

修改群群公告传roomNotice参数，roomName传""

都修改，则都传

代码

```
IMManager.getInstance().updateRoomBaseInfo(roomId, roomName, roomNotice, new
ResponseCallBack<String>() {
    @Override
    public void onSuccess(String result) {
    }

    @Override
    public void onFailure(String result) {
    }
});
```

17.设置群信息

修改群名称传roomName参数，roomNotice传""

修改群群公告传roomNotice参数，roomName传""

都修改，则都传

代码

```
int type: 0:显示群消息已读人数 1:组是否公开 2:加群是否需要验证 3:允许成员邀请好友 4:允许成员
上传群共享文件
boolean isChecked:
type=0 true 显示 false 关闭
type=1 true 不公开 false 公开群组
type=2 true 开启验证 false 关闭验证
type=3 true 允许 false 禁止
type=4 true允许 false 禁止
```java
IMManager.getInstance().updateRoomInfo(roomId, type, isChecked, new
ResponseCallBack<String>() {
 @Override
 public void onSuccess(String result) {
 }

 @Override
 public void onFailure(String result) {
 }
});
```

## 18.指定管理员

toUserId: 指定管理员用户 id

### 代码

```
IMManager.getInstance().setRoomManager(roomId, toUserId, new ResponseCallBack<String>
() {
 @Override
 public void onSuccess(String result) {
 }

 @Override
 public void onFailure(String result) {
 }
});
```

## 19.删除群公告

noticeId: 公告 id



## 代码

```
IMManager.getInstance().deleteNotice(roomId, noticeId, new ResponseCallBack<String>()
{
 @Override
 public void onSuccess(String result) {
 }

 @Override
 public void onFailure(String result) {
 }
});
```

## 20.设置消息免打扰

int type 0:关闭, 1: 开启

## 代码

```
IMManager.getInstance().updateDisturbState(roomId, type, new ResponseCallBack<String>
() {
 @Override
 public void onSuccess(String result) {
 }

 @Override
 public void onFailure(String result) {
 }
});
```

## 21.禁言

## 代码

```
IMManager.getInstance().shutUpMember(roomId, userId, true, new
ResponseCallBack<String>() {
 @Override
 public void onSuccess(String result) {
 }

 @Override
 public void onFailure(String result) {
 }
});
```

## 22.拉黑

## 代码

```
IMManager.getInstance().addBlacklist(userId, new ResponseCallBack<String>() {
 @Override
 public void onSuccess(String result) {
 }

 @Override
 public void onFailure(String result) {
 }
});
```

## 23.取消拉黑

### 代码

```
IMManager.getInstance().removeBlacklist(userId, new ResponseCallBack<String>() {
 @Override
 public void onSuccess(String result) {
 }

 @Override
 public void onFailure(String result) {
 }
});
```

## 24.好友列表

### 代码

```
IMManager.getInstance().getFriendList(new ResponseCallBack<List<AttentionUser>>() {
 @Override
 public void onSuccess(List<AttentionUser> attentionUsers) {
 }

 @Override
 public void onFailure(String result) {
 }
});
```

## 25.添加好友

## 代码

```
IMManager.getInstance().addFriend(userId, new ResponseCallBack<String>() {
 @Override
 public void onSuccess(String result) {
 }

 @Override
 public void onFailure(String result) {
 }
});
```

## 26.同意好友申请

### 代码

```
IMManager.getInstance().agreeAddFriend(userId, new ResponseCallBack<String>() {
 @Override
 public void onSuccess(String result) {
 }

 @Override
 public void onFailure(String result) {
 }
});
```

## 27.删除好友

### 代码

```
IMManager.getInstance().deleteFriend(userId, new ResponseCallBack<String>() {
 @Override
 public void onSuccess(String result) {
 }

 @Override
 public void onFailure(String result) {
 }
});
```

## 28.设置推送凭证

pushType 推送类型 使用 PushType枚举中的类型

voucher 推送凭证

## 代码

```
IMManager.getInstance().setPushVoucher(pushType, voucher, new ResponseCallBack<String>
() {
 @Override
 public void onSuccess(String result) {
 }

 @Override
 public void onFailure(String result) {
 }
});
```