

聊天 App 中实时通信设计与实现

罗泓汶 杨声英

四川大学锦城学院 计算机与软件学院 四川 成都 611731

【摘要】：在二十一世纪的今天，随着网络快速发展，出现了很多的网络社交聊天软件，比如国外的 Facebook、ins，以及国内腾讯的 QQ 和微信等。但是这些软件的功能都是大同小异，关键就在于实时通信，把消息即时反馈给用户。本文基于 Android 平台，简述了第三方 IM 实时通讯方案的实现步骤，并利用 WebSocket 协议实现实时通信功能，以解决当第三方 IM 实时通讯方案无法使用的情况。

【关键词】：实时通信；聊天软件；Java-WebSocket 框架

如今，在互联网时代的不断发展下，世界的信息技术逐渐完善，我国也紧随其后慢慢成为世界最大的互联网用户国家。人们在网络世界饥渴的探索、不断结交陌生的朋友、在私有的信息库中储存自己的美好回忆，以及分享自己的生活趣事，而需要完成这些事件的工具就是实时通信软件。在 2008 年之前的中国，互联网共享信息是不完善的，所以人们通过电子设备交流的方式还局限于电话和短信。但随着互联网时代的开发，实时通信软件体现出重要的地位，这种软件逐渐替代了电话和短信。它不同于电话和 E-mail，实时通信软件的交谈方式是实时的，在你给对方发送信息完毕的时间点上，对方会立即收到你所发送的信息，通信过程中的消息记录也会保存在用户的账号中方便查阅。现在 Android 系统成为了手机主流系统之一，对比其他系统而言 Android 系统具有开放灵活的优势，所以更简单容易的实现实时通信。

1 第三方 IM 实时通讯方案分析

实时通讯最显著的特点就是高接通率、高可靠性，不能有明显的推送延迟。在通信技术的不断发展，已经出现了很多第三方的 IM 实时通讯方案。市面上常见的有：融云、JPush、网易云信、环信、云旺 OpenIM、BigAnt。这些方案给开发人员带来了很大便捷，不用再去开发出属于自己的一套通讯方案。因为第三方实时通讯方案现在有很多，所以应该从各方面考虑选用适合自己的方案。例如：稳定性、功能性、安全性、运维服务、SDK、协议和费用。在了解熟悉这些方面后再去对比各个不同的第三方。

1.1 第三方通信方案简介与实现

融云 IM 有两个版本：商用版和免费版。它现在每日支持免费消息超过了 1000 条，还能够满足各类通信环境和互联网业务的基本需求，但是如果超过免费消息数量或者需要 IM 的海外节点服务就要支付一定的费用。

JPush 极光推送相比较融云，JPush 的日推消息数比融云

更多，每日更是超过了五亿条。JPush 不仅可以全方面支持 Android、iOS、Winphone 等手机平台，还可以通过 API 推送消息，同时 JPush 极光推送可以提供可视化的 web 端控制台发送通知和推送效果的统计分析。

网易云信可以让开发者通过集成客户端 SDK 和运端开放 API，迅速的实现了互联网通信和音视频等强大的功能。网易云信在场景化方面更是做的极致，帮助企业完善了行业中的难题。同时移动端的互踢（ios 和 Android）、桌面端的互踢（PC 和 web）以及移动端和桌面端的共存是网易云信内置策略。如果企业的业务无法用互踢策略满足，就可以申请取消，从而再根据多端登录回调来判断当前设备是否需要被踢出。

BigAnt（大蚂蚁）是我国企业自主研发的实时通信方案以及软件领导品牌，它是我国第一家电子政务实时通讯软件。它的工作领域是企事业单位和政府部门。它通过对政府企业机构中内部的现有信息和应用体系快速实现企业内部的实时通信、局域网文件共享管理、组织架构分权限管理、一站式的协同办公。它可以大大的有效缩短内部沟通距离以来提高企业内部工作运作效率。以下我们以融云推送为例，简单说明其实现过程。

1.2 融云推送的简单实现

（1）环境准备

当使用融云 IM 作为第三方通信方案时，首先需要在融云官网注册一个开发者账号，并下载融云的 SDK，这样就可以根据自己开发需求来下载各个组件。如果简单的实现聊天就使用 IMKit（界面组件）和 IMLib（通讯能力库）即可。之后集成 SDK，导入对应 Moudle 的代码，并在融云获取两套 AppKey/Secret 以避免数据冲突，每个 App/包名对应一套 Key/Secret。

完成以上操作后，在开发界面先打开 Module 的 build.gradle，在 dependencies 类中添加对应模块的依赖，然后

打开 AndroidManifest.xml 文件配置 Appkey, 设置好 name 和 value, 继续在 Manifest 文件中配置 FileProvider。

(2) 获取授权

继以上步骤以后, 需要在 APP 中先完成初始化 SDK。下一步需要获取 Token 用来保证身份授权和安全, 并在获取过程中对参数签名, 调节 connect()方法连接服务器, 注意在方法中写出 log 日志以来用于判断是否连接成功。为了实现接受消息, 需要自定义一个继承自 PushMessageReceiver 类的 BroadcastReceiverI (广播接收器) 类, 实现自定义类中的 MessageArrived 与 MessageClicked, 然后把 receiver 注册到 Manifest 文件中。

(3) 界面配置

完成以上步骤之后, 需要去设计聊天模块的 UI 界面、配置联系人 Activity、配置会话界面 Activity。配置会话界面可以使用融云的 Fragment 即可。最后在 onCreateView 方法中配置会话记录 Fragment, 因为只有有了消息记录才会显示。

(4) 消息设置

最后就是发送消息的设置, 先构造 TextMessage 实例, 生成 Message 对象, 然后通过 IRongCallback.ISendMessageCallback 中的方法回调消息状态和消息体。其中需要设置 message 是将要发送的消息体, 如果发送的是自定义消息必须填写字段。完成后我们就可以使用发送消息基本功能。

2 WebSocket 实现实时通信

在实现实时通信时, 虽然有了第三方 IM 通信方案, 但是如果项目有了特殊要求不能使用第三方, 就需要自己重新开发了, 比如可以使用 WebSocket 来完成开发工作。首先要了解 WebSocket 协议, WebSocket 协议是一个可以建立长连接的全双工的通信协议, 它还可以允许服务器端发送讯息给客户端。如果我们想在 Android 端使用 websocket 协议, 可以选用 Java-WebSocket 框架, 选择的理由是: 第一这个框架是开源免费的, 其次现在它还在不断地进行维护更新。

通过引入 Java-WebSocket 框架实现 WebSocket 实现实时通信步骤如下:

(1) 初始化客户端

首先要引入 Java-WebSocket 框架, 在 build.gradle 中加入 “org.java-websocket:Java-WebSocket:1.4.2”, 再加入网络的请求权限, 其次新建一个继承 WebSocketClient 的客户端, 并且实现客户端里面的 onOpen、onMessage、onClose、onError

这四个抽象方法和构造函数。onOpen()方法在 websocket 连接时候开启调用, 接收消息时候调用 onMessage()方法, 断开连接时候调用 onClose()方法, 连接出错时候调用 onError()方法。

其次需要初始化客户端再调用连接方法以便建立 websocket 的连接, 注意: 1) 创造对象不能重复使用, 2) 初始化客户端时候需要传入 websocket 的地址。连接时候可以选着使用 connect()方法和 connectBlocking()方法, 需要注意的是 connectBlocking()方法会先连接再发送, 这样就会多出一个等待操作。运行之后我们就可以发现客户端上执行了 onOpen()方法, 这就表示已经与 websocket 建立了连接。这样就可以调用 send()方法来发送消息, 再调用 close()方法关闭 websocket 连接时候需要将对象设置为 NULL, 为了避免重复实例化。

(2) 服务端设置

首先新建一个 Service, 在启动 Service 时候让实例化 WebSocketClient 对象并且建立连接。其次需要获取到 Service 中 WebSocketClient 对象即用到 Service 中的 onBind()方法。先创建一个继承 Binder 的 Binder 类, 在 onBind()方法中返回这个类的实例。接下来需要做的是在对应的 Activity 绑定 Service, 并获取 Service 里的内容。可以先建一个 ServiceConnection 匿名类, 在匿名类里面重写 Connected()和 Disconnected(), 在活动与服务器成功绑定或者断开时候会调用这两个方法。在 Connected()中会先获取 ClientBinder 实例, 因为在获取到这个实例后我们就可以调用服务端的 public 方法。之后我们需要调用 getService()中的 Service 实例, 从而获取到 WebSocketClient 对象, 也就是可以在活动中推送讯息了。

(3) 更新 UI

通过广播来实现在 Service 中更新 Activity 的 UI, 这里和第三方的通信解决方案相似。在对应 Activity 中自定义广播接收者, 然后在 Service 中获取到消息发出的广播即可。

(4) 消息通知

完成 UI 配置以后, 需要完成的是消息通知, 在接收到消息时候会有通知弹出。消息通知可以直接使用 Notification, 首先确定好功能的需求即设置通知的优先级、通知的声音、闪光灯和震动效果。如果我们为收到通知, 先要确定通知的设置是否打开, 如果是锁屏时候无法弹出通知, 我们就要先获取锁屏通知权限。

(5) 系统检测和服务器保活

最后需要对应用进行心跳检测和重连，这是为了保证 websocket 的稳定连接。其实在 Android 7.0 版本以后就没有什么真正意义上的保活，但是一个应用还是需要做一些处理和维 护。保活的方式有很多种，例如一像素保活、进程互拉、申请自启权限等等。

3 结语

Android 平台本身的开源优势给开发者带来了更多的发

挥空间，在新技术不断的引领下，聊天 APP 的功能逐渐完善，再给用户带来便利的时候同时也出现了新的需求。实时通信的技术也不断运用到我们的生活中，小到发送一条讯息，大到多人视频共享会议，甚至以后在 5G 下我们可以实现全息投影会议。今年的疫情爆发在给经济上造成重创时，也让通信技术面临挑战，孩子需要在家上网课、父母需要在家办公、医生需要远程协助等。但是我们要有足够的信心去克服难题。开发者更是需要拥有一颗探索精神。

参考文献：

- [1] 于中天.基于 Android 的校园聊天软件的设计与实现[J].电脑迷,2017(09):55.
- [2] 郝胜华,张素华.基于 JAVA 的聊天软件设计与实现[J].电脑知识与技术,2017,13(14):97-99.
- [3] 万芳君.即时聊天软件设计与应用[J].电脑迷,2017(01):25+27.