**一、即时通讯**

**什么是即时通讯？**

* 即时通讯，又称实时通讯
* 即时通信（Instant Messaging，简称IM）是一个实时通信系统，允许两人或多人使用网络实时的传递文字消息、文件、语音与视频交流
* 即时通讯在开发中使用的场景
  + 开发一个类型于微信，QQ，易信的聊天软件
  + 在电商APP集成买家与卖家的实时沟通等
* 当前时下，实现即时通讯的方案
  + 1.XMPP
  + 2.环信

**二、XMPP**

**XMPP是什么？**

* (1).XMPP：The Extensible Messaging and Presence  
  Protocol（可扩展通讯和表示协议）
* (2).XMPP是一种基于XML的即时通讯协议，XMPP的官方文档是RFC 3920
  + 这个文档定义了登录，退出，获取好友，发送消息等等的XML数据传输协议
* (3).XMPP是一个典型的C/S架构，基本的网络形式是客户端通过TCP/IP连接到服务器，通过Socket建立连接，然后在之上传输XML流
* (4).XMPP是一种类似于HTTP协议的一种数据传输协议，其过程就如同“解包装 --〉包装”的过程。只需要理解其接收的类型及返回的类型，便可以很好的利用XMPP来进行数据通讯
* (5).XMPP官方网站——<http://xmpp.org>

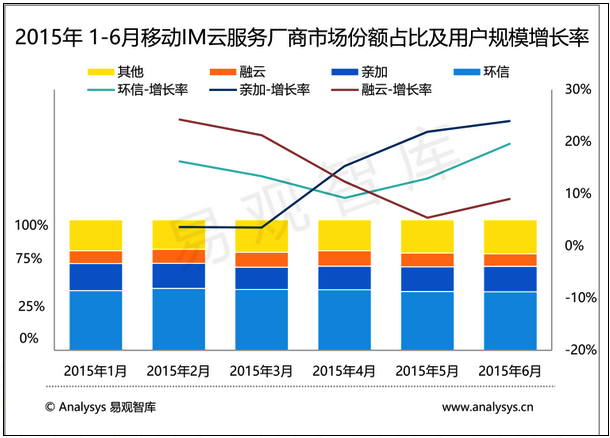
**XMPP实现即时通信的准备工作**

1-下载Openfire服务器  
2-下载XMPPFramework框架

**三、环信**

**什么是环信？**

* 环信是一个第三平台，提供即时通信(IM –Instant Messaging )的服务
* 环信是在XMPP的基础上进行二次开发
* 环信在网络上传输的数据也是XML
* 使用环信，不用自己搭建服务器，节约成本
* 环信日活30万以下,永远免费
* 2015年上半年易观发布中国移动IM云报告，环信稳居市场第一
* 公司如要开发即时通讯软件，建议首选环信,环信占用市场份额较大



各类IM服务占比例图的变化.png

**集成环信的前提准备**

* 注册成为环信开发者  
  <http://www.easemob.com>
* 在开发者后台创建APP获取Key
* 官方SDKDEMO  
  <http://www.easemob.com/docs/ios/IOSSDKPrepare/>



集成环信的前提准备.png

* 根据官网导入SDK和相应依赖
* 初始化应用,有两个方法

/\*

\*registerSDKWithAppKey:区别app的标识，开发者注册及管理后台apnsCertName:iOS中推送证书名称。制作与上传推送证书

\*/

//环信的初始化

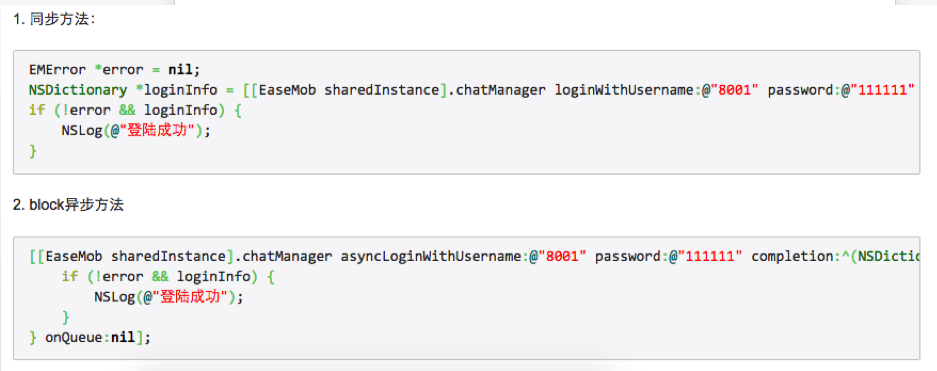
// [[EaseMob sharedInstance] registerSDKWithAppKey:@"MG#MGChat"

ap sCertName:@””];

//环信的初始化

并隐藏日志输出

[[EaseMob sharedInstance] registerSDKWithAppKey:@"MG#MGChat" apnsCertName:@"" otherConfig:@{kSDKConfigEnableConsoleLogger:@(NO)}];



登录.png

* 注册
* [[EaseMob
* sharedInstance].chatManager asyncRegisterNewAccount:”MG” password:”123456” withCompletion:^(NSString \*username, NSString \*password,EMError \*eror) {
* NSLog(@"error:%@,username:%@,pwd:%@",error,username,password);

} onQueue:nil];

* 自动登录
  + 自动登录：当然程序再次启动时，如果用户已经成功登录过，不需要用户再输入用户名和密码进行登录
  + 实现方法  
    (1)把登录帐号信息保存下来，程序起动发送登录请求  
    (2)环信已经实现了自动登录，只需要在第一次登录成功后调用 下面的方法即可  
    [[EaseMobsharedInstance].chatManagersetIsAutoLoginEnabled:YES];
* 总结：不管哪种方法，实质都是再次调用了登录的网络请求
* 掉线自动登录  
  -如果网络不通过，用户应该自动连接到服务器,以及时接收消息
  + 此功能无需程序员自己做，环信框架已实现,环信SDK会调用自动连接的代理方法来通知应用程序

/\*!

@method

@brief 将要发起自动重连操作时发送该回调

@discussion

@result

\*/

- (void)willAutoReconnect;

/\*!

@method

@brief 自动重连操作完成后的回调（成功的话，error为nil，失败的话，查看error的错误信息）

@discussion

@result

\*/

- (void)didAutoReconnectFinishedWithError:(NSError\*)error;

* 好友
  + 上面的协议的实现了对用户的基本操作，如  
    (1)添加好友  
    (2)从本地获取好友列表  
    (3)从服务器获取最新好友列表  
    (4)接收好友添加请求  
    (5)删除好友  
    (6)被好友从名单上删除
* 聊天  
  环信消息发送的流程  
  1.先把记录保存到Conversation表  
  2.接着发送网络请求，API如下
* [[EaseMob sharedInstance].chatManager asyncSendMessage:message progress:self prepare:^(EMMessage \*message, EMError \*error) {
* NSLog(@"prepare %@",message.messageBodies);
* } onQueue:nil completion:^(EMMessage \*message, EMError \*error) {
* NSLog(@"完成%@",message.messageBodies);

} onQueue:nil];

* + 步骤  
    ①先获取EMConversation会话管理者对象  
    ②调用EMConversation的  
    - (NSArray\*)loadNumbersOfMessages:(NSUInteger) aCountbefore:(long long)timestamp;获取指定的聊天记录
* 环信提供会话管理者(EMConversation)来管理未读消息数和历史聊天记录,具体代码如下
* 总的未读消息数需要遍历conversations

// 1.获取所有历史会话

NSArray \*conversations = [[EaseMob sharedInstance].chatManager conversations];

// 2.如果内存中，没有会话，从数据库中加载

if(conversations.count == 0)

{

conversations = [[EaseMob sharedInstance].chatManager loadAllConversationsFromDatabaseWithAppend2Chat:YES];

}



Snip20160421\_6.png

* 当进入聊天页面时，需要设置所有当前会话信息或者设置已经加载的消息为已读

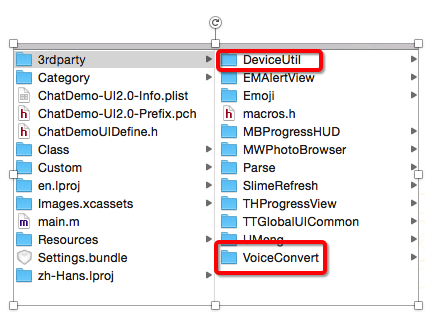
// 设置当前会话所有消息都为已读

[self.conversation markAllMessagesAsRead:YES];

//设置某条消息为已读

[self.conversation markMessageWithId:<#(NSString \*)#> asRead:<#(BOOL)#>]

* 语言消息
  + 在录音前导入环信封装的两个录音框架，如图



导入录音的框架.png

* + 刚才导入的两个框架，已经实现了录音Api
  + // 开始录音
  + [[EMCDDeviceManager
  + sharedInstance]
  + asyncStartRecordingWithFileName:fileName completion:^(NSError \*error){
  + if(error) {
  + NSLog(@"failure to start recording");
  + }
  + }];
  + // 结束录音
  + [[EMCDDeviceManager sharedInstance] asyncStopRecordingWithCompletion:^(NSString \*recordPath, NSInteger aDuration, NSError \*error) {
  + NSLog(@"%@",recordPath);

}];

// 语音对象

EMChatVoice \*voice = [[EMChatVoice alloc] initWithFile:filePath displayName:@"audio"];

//消息体

EMVoiceMessageBody \*body = [[EMVoiceMessageBody alloc] initWithChatObject:voice];

EMMessage \*message = [[EMMessage alloc] initWithReceiver:self.buddy.username bodies:@[body]];

message.messageType = eMessageTypeChat;// 私聊

// 不加密

message.requireEncryption = NO;

// 播放完成

[[EMCDDeviceManager sharedInstance] asyncPlayingWithPath:filePath completion:^(NSError \*error) {

NSLog(@"播放完成%@",error);

}];

* 退出(异步方法)
* [[EaseMobsharedInstance].chatManagerasyncLogoffWithUnbindDeviceToken:YEScompletion:^(NSDictionary \*info, EMError \*error) {
* if (!error) {//退出成功
* }else{//退出失败;
* }
* }

onQueue:nil];

**补充三点总结：**

**使用XMPP遇到的问题**

* 发送附件（图片，语音，文档…）时比较麻烦
* XMPP框架没有提供发送附件的功能，需要自己实现
  + 实现方法，把文件上传到文件服务器，上传成功后获取文件保存路径，再把附件的路径发送给好友

**XMPP的实现原理**

* XMPP是一个即时通讯的协议，它规范了用于即时通信在网络上数据传输格式的，比如登录，获取好友列表等等的格式。XMPP在网络传输的数据是XML格式
  + 比如登录：把用户名和密码放在xml的标签中，传输到服务器
* XMPP是一个基于个Socket通过的网络协议，目的是为了保存长连接，以实现即时通讯功能
* XMPP的客户端是使用一个XMPPFramework框架实现
* XMPP的服务器是使用Openfire,一个开源的服务器
* 客户端获取到服务器发送过来的好友消息，客户端需要对XML进行解析，使用的解析框架的KissXML框架，而不是NSXMLParser/GDataXML

**简单的说下环信的实现原理**

* 环信是一个即时通信的服务提供商
* 环信使用的是XMPP协议，它是再XMPP的基础上进行二次开发，对服务器Openfire和客户端进行功能模型的添加和客户端SDK的封装，环信的本质还是使用XMPP，基本于Socket的网络通信
* 环信内部实现了数据缓存，会把聊天记录添加到数据库，把附件(如音频文件，图片文件)下载到本地，使程序员更多时间是花到用户即时体验上
* 环信内部已经实现了视频，音频，图片，其它附件发送功能
* 环信使用公司可以节约时间成本
  + 不需要公司内部搭建服务器
  + 客户端的开发，使用环信SDK比使用XMPPFramework更简洁方便

作者：Mg明明就是你  
链接：http://www.jianshu.com/p/e5dd0d712996  
來源：简书  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。