**TeamTalk-web使用说明**

TeamTalk-web项目包含服务端和客户端两部分，其中服务端使用nodejs，客户端使用JavaScript。

使用步骤：

1. 操作系统为Windows；
2. 安装[nodejs](https://nodejs.org/en/download/)和Tomcat或者nginx（获取其他Web服务器都可）；
3. 修改TeamTalk-web根目录下的tt2http.js文件中TeamTalk的IP地址，更新为你们自己的TeamTalk服务器IP地址即可，修改位置具体在第38行；
4. 使用命令行“node tt2http.js”运行tt2http.js，将启动一个用于和TeamTalk服务器通信的TCP服务和一个用于和Web端通信的WebSocket服务；
5. 将tt-client文件夹中的所有文件拷贝到Web服务器工程目录中发布，我使用的是Tomcat，简单说明一下如何使用：
   1. 将tt-client整个文件夹拷贝到Tomcat的webapps/ROOT目录下；
   2. 使用命令行进入到Tomcat/bin目录下，运行“startup.bat”启动tomcat；
   3. 浏览器中输入<http://127.0.0.1:8080/tt-client>即可访问TeamTalk Web端，浏览器推荐使用Chrome或者Firefox，IE也许会存在兼容性问题；
6. 使用TeamTalk Web前还需要修改index.html文件中Web Socket的ip地址，修改为运行步骤4中主机IP地址，具体在第132行；
7. 打开TeamTalk Web界面后，输入已在TeamTalk后台创建好的用户名密码，点击“Login”进行登录；
8. 登录成功后，左下方的“Buddies”里面会显示好友列表；
9. 使用鼠标在任一好友名上单击，即可开始给该好友发送消息；
10. 点击好友名后，在右侧的输入框中开始输入，点击“Send”按钮即可给好友发送消息；

**Tips** 注意上述步骤3和步骤6中的IP地址修改

tt-client简单介绍：

1. 文件夹中的文件结构介绍：
   1. index.html文件为TeamTalk Web的主界面，做界面的猿们主要看这个文件；
   2. ByteBufferAB.min.js、Long.min.js、md5.js、ProtoBuf.min.js和tt-message-pb文件夹为PB相关的，只管引用不用看里面的具体内容；
   3. zepto.js为JS框架文件，可以换成其他的，JQuery什么的；
   4. 重点是tt2http-client.js这个文件，WebSocket的客户端，提供TeamTalk消息的封装、解析和回调方法。
2. TeamTalk Web端相关JS的具体使用方法：
   1. 主要对照index.html文件来进行讲解；
   2. 和index.html文件一样，在文件头引用ByteBufferAB.min.js、Long.min.js、ProtoBuf.min.js、和tt2http-client.js（zepto.js可以换成其他JS框架文件）；
   3. tt2http-client.js提供了一个TTWebClient类，如index.html文件132行所示，构造TTWebClient时需要提供5个参数：1.运行tt2http.js服务的主机IP地址，2. tt2http.js监听的端口号，3.接收消息的回调函数(第88行)，4.与WebSocket服务端连接状态变化处理的回调函数(第107行)，5.当前登录用户名；
   4. 调用TTWebClient.on()注册其他回调函数，该方法第一个参数为回调函数处理的消息类型，第二个参数为回调函数名。我们暂时只提供了“buddy\_list”相关消息的回调处理，之后再新增其他消息的回调处理时，注册回调函数的方式和这个一致，可以多次调用TTWebClient.on()注册多个回调函数。
   5. 发送消息是直接调用TTWebClient.sendMessage()，参考第81行，其中第一个参数是消息接收方用户的ID，第二个参数是消息内容；
   6. 目前提供的“buddy\_list”消息的回调处理是用于接收到我的好友列表后的消息，回调函数提供一个包含好友信息的数组类型参数，参考第92行onBuddiesUpdated()方法；