

基于Nodejs全开放式匿名交流平台的搭建

李锦涛

临邑县第一中学, 山东德州 251500

摘要 21世纪是信息的世纪, 交流便是获取信息最简单而快捷的方式。本文旨在解决在初次见面、匿名沟通等场合下, 人们之间交流难以畅所欲言的情景。本文以 Node.js 技术搭建了一个全开放式的匿名聊天交流平台。其中, 采用 express 技术搭建 Web 服务器, 任何用户通过浏览器向后台发起 http 请求, 便能接收到一个大幕布, 大幕布可匿名呈现所有用户的相关聊天信息; 采用微信公众号作为中转站, 所有关注该特定公众号并向该公众号发送的信息, 均由微信后台服务器转到笔者搭建的后台服务器, 然后再通过 WebSocket 技术广播到所有的大幕布中。如此, 人们之间的交流方式既有了匿名的保障, 同时又开放式的呈现所有用户的聊天内容, 从而达到良好沟通、表达个人意愿的目的。

关键词 WebSocket; node.js; 微信墙; 交流平台

中图分类号 TP3

文献标识码 A

文章编号 1674-6708 (2018) 204-0153-03

随着科技的飞速发展, 人们的通讯方式也变得更加便捷。根据美国市场研究公司 IDC 的报告, 2017 年全球智能手机出货量达到 15.2 亿部, 而中国智能手机的出货量达到 4.6 亿部, 针对年轻人群来说, 几乎人人手持一部智能手机。由于手机实时通讯软件的便利, 人们交换信息的方式更多的是选择在手机端完成。

在中国, 腾讯的两款明星产品, 微信和 QQ 已经霸占了市场上实时通讯软件的老大位置。据腾讯官网报道, 2017 年微信的月活跃账户数到 9.63 亿, QQ 的月活跃账户数则说 8.50 亿。人们智能手机上几乎都装有这两款实时通讯软件, 每天与认识的朋友和不太熟悉的朋友进行信息之间的交流。

然而, 微信与 QQ 这两款流行的实时通讯软件在众多场合下并不具备信息的匿名和信息的全开放式, 人们之间的交流受到或多或少的局限。在信息的匿名方面上, 例如在班级选举上, 采用同学建 QQ 群或者微信群, 然后大家在群里进行讨论、交流、投票。这显然不是一个特别合适的举动, 因为碍于情面, 同学们大都回避以公开的身份来讨论谁的优缺点, 以及是不是该投他一票; 又比如, 在新生见面交流会, 或者班级讨论会, 一些内向的学生在新的环境下, 很难站出来表达自己

的观点。如果, 能够在开放式的平台下, 选择匿名的方式畅所欲言, 对它们而言或许是一个更好的选择。此外, 在信息的全开放式方面上, 例如家人想了解一下自己的孩子在生活学校生活一些讨论会, 或者某些公开场合下的交流情况的时候, 不太可能让家人加入各种微信群或者 QQ 群, 那么家人就无法获取同学们的聊天信息。针对这种公开的场合和活动, 如果能过让众多家长获取孩子在学习的表现, 不仅增加了长备与孩子之间的话题, 也让家人对孩子的成才给点自己的建议。

为了解决目前市面上实时通讯系统在匿名方式和全开放式方面上, 人们交流方式潜在的不适应性, 本文基于 Node.js 搭建一个全开放式匿名交流的平台, 在特定场合下, 全方面的满足人们之间获取信息、交换信息的便利性。其中, 本系统应该具备以下两个主要特点: 1) 对所有用户全方位开放: 任何用户可以随时在平台上获取当前所有用户讨论的最新消息; 2) 对发布信息的所有者实时匿名, 以保障个人信息不被泄露。

1 系统总体框架设计

本文采用 Node.js^[1-2] 搭建了后台服务器, 其中 Web 端采用 express 框架编写。如图 1 所示, 用户通过智能手机 (Android 系统或者苹果系统) 关注某个特定微信公众号来与后台服务器绑定,

作者简介: 李锦涛, 临邑县第一中学。

微信后台服务器作为用户与我搭建的后台服务器的中转桥梁。具体通讯流程如下：

首先，任何用户可以在任何有网络资源的地方通过浏览器访问后台服务器（向服务器 118.190.27.173:8000 发送 http 请求，其中端口号为 8000）；此时，后台服务器返回给用户一个大幕布，即 html 页面，该页面包含了后台服务器进行 Websocket 通信的代码。

其次，进行微信公众号与后台服务器进行绑定绑定，即往该特定公众号发送的所有内容均被微信公众号服务器所管理，然后再转给我自己搭建的后台服务器。

最后，所有关注该特定微信公众号的用户，往微信公众号发送信息，均间接转到搭建的后台服务器，然后通过 Websocket 的方式，广播给所有打开大幕布的用户。

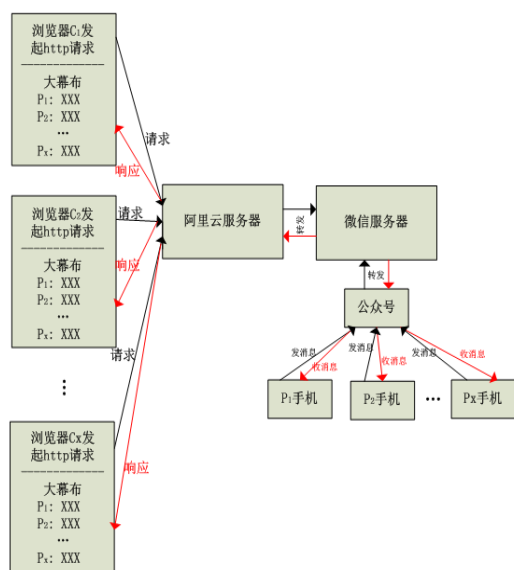


图1 全开放式匿名交流平台的系统框架图

2 主要模块介绍

2.1 express 模块

Express^[3] 模块是一个基于 Nodejs 平台的简化版 Web 应用开发框架，可以用短短几行代码搭建一个简易版的 Web 平台，可以影响 http 的请求，并返回给浏览器一个 html 的页面，在本文中，则是返回一个大幕布，可以匿名呈现用户者的聊天信息。其中，在 Nodejs 中采用 npm 包管理模块来统一管理开源的第三方包。因此，可以采用 npm 来下载 express 相应的包，命令如下：

npm install express--save 和 npm install express-generator

用了短短 9 行的代码，便成功搭建了一个迷你型的 web 后台服务器。其中，代码概要解释如下：

```

1) var express = require('express');
2) var app = express();
3) app.use(express.static(__dirname + '/client'));
4) app.use(function (req, res) {
5)   res.sendFile(__dirname + '/client/index.html');
6) }
7) var server = app.listen(8000, function () {
8)   console.log('app is running at port 8000!');
9) });

```

第 1 行，引入刚刚才有 npm 包管理器下载好的 express 包；

第 2 行，实例化 express 对象，并命名为 app 变量；

第 3 行，挂载本地静态文件目录，其中大幕布 html 文件就放置在其目录下；

第 4~5 行，挂载一个匿名函数，该函数监听客户端的 http 请求，若用户在浏览器发起 http 请求，则调用该函数，并完成将本地静态文件路径下的大幕布 html 文件（即 index.html）返回给用户。此时，用户可以收到一个干净的大幕布，里面还没有显示用户者的聊天内容。

第 6~7 行：开启 Web 服务器的监听，监听的端口号为 8000。因为该服务器的 IP 地址为 118.190.27.173，Web 监听的端口号为 8000。因此，如果用户在浏览器发起 http://118.190.27.173:8000 的请求，Web 服务器便能实时响应。

2.2 Websocket

Websocket^[4-5] 是一种应用层协议，旨在为 web 应用程序客户端和服务端之间提供一种全双工的通信机制。本项目中，由大幕布 html 网页向后台服务器发起 websocket 通信，具体代码如下：

```

var ws = new WebSocket('ws://118.190.27.173:9530');
ws.onmessage = function(event) {
  //do something
}

```

其中，采用 9530 端口号与后台服务器进行通信，ws 为返回的对象。然后 ws 通过调用 onmessage 函数来监听后台服务器是否发生广播事件（服务器后台 wss.Broadcast(data) 来进行广播），若收到广播的数据，则数据保存在 event 变量中，然后进行后续数据的处理。本项目中数据保护用户的微信人头像和用户往特定微信公众号发送的内容，经过简单后续处理，以倒序的形式显示在大幕布中。

3 项目效果展示

首先，在浏览器中发起 `http://118.190.27.173:8000` 请求，此时后台服务器监听到 8000 端口有用户发来的 http 请求，便响应客户端的请求，返回一个 html 页面，即一个空白的大幕布。

接着，同时关注了 yezhenxu001 的接口测试号的用户 A 和用户 B 开始往公众号发送消息，具体内容如图 2 所示，此时微笑公众号后台将消息转发给我自己搭建的后台服务器。值得注意的是，用户 A 与用户 B 是互相不认识的，而且没有互相添加为好友。



图2 用户A与用户B真实的聊天对话图

最后，大幕布可以实时更新呈现用户之间的所有当前聊天内容，具体如图 3 所示。可以看出，用户 A 和用户 B 的聊天记录均在大幕布中所呈现，而去用户的个人信息得到了匿名，只能看到个人微信头像。当然如果想进一步提高匿名的程度，可以把个人微信头像给隐藏掉。如此一来，一方面，用户的个人隐私得到了有效的保障；另一方面，所有想关注用户之间的相关聊天信息，可以通过向后台服务器发起 http 请求，得到大幕布，然后展示用户者的当前聊天记录。



图3 大幕布实时呈现用户者之间的所有聊天信息

4 结论

文章分析了当前主流即时通信软件微信和 QQ 在某些特定场合下不适应的情况，并针对该场景设计了一款全开放式匿名交流平台。在实际应用中，不仅方便了用户之间匿名交流的需求，同时也极大地满足了对该聊天内容感兴趣的群体。因此，该平台在用户交流需求中具有较大的实际应用价值。

参考文献

- [1]程桂花, 沈炜, 何松林, 等. Node.js 中 Express 框架路由机制的研究[J]. 工业控制计算机, 2016 (8): 101-102.
- [2]崔莹, 刘兵. Node.js 与 Express 技术在计算机课程教学中的应用[J]. 软件导刊, 2016, 15 (9): 190-192.
- [3]<http://www.expressjs.com.cn/> express 官方网站.
- [4]单振华, 王舒憬, 强杰. 基于 WebSocket 的智能家居远程控制系统的实现[J]. 电子技术应用, 2017, 43 (10): 124-127.
- [5]<http://www.52im.net/thread-331-1-1.html> “WebSocket 详解 (一): 初步认识 WebSocket 技术”.