课程大纲

- ◆课程概述(什么是消息推送?有哪些应用场景?)
- ◆H5 原生WebSocket快速入门(实现流程、原理,异常重连、心跳检测)
- ◆SockJS+Stomp实现WebSocket(解决兼容性问题和简化开发)
- ◆消息队列作为Stomp消息代理(RabbitMQ作为消息代理案例)
- ◆消息推送负载均衡实现(难点、Nginx方案、SpringCloudGateway方案)
- ◆更多章节,更新中...

课程特色

- ◆由浅入深,从h5WebSocket原生开始→SockJS→Stomp→RabbitMQ Stomp→Nginx负载均衡→SpringCloudGateway负载均衡
- ◆聚焦于WebSocket消息推送,我们只要真正掌握了消息推送,就能在各种业务系 统中运用自如
- ◆动手敲代码机会多,80%的课时都有源码,大家既可以自己手动敲代码,也可以用本人的示例代码跑效果,还可以在此基础上修改、拓展
- ◆流程图很多,一边讲流程图,一边敲代码实现,不断掌握编码技巧,还能深入理解原理、流程

适合人群

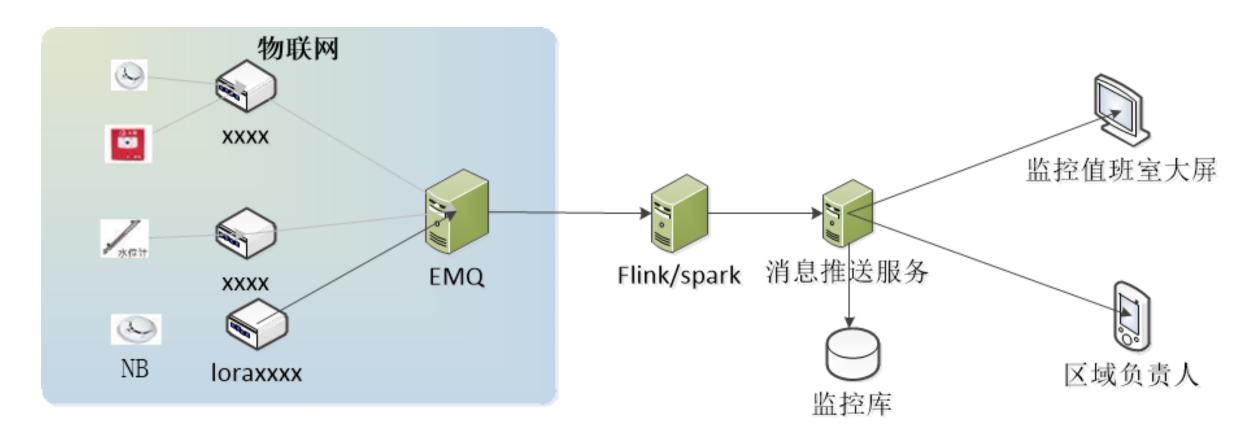
- ◆学习者要求
 - ◆熟悉java开发语言
 - ◆有spring boot或spring mvc基础
- ◆适合人群
 - ◆有一定基础Java开发人员
 - ◆职场充电

学习收获

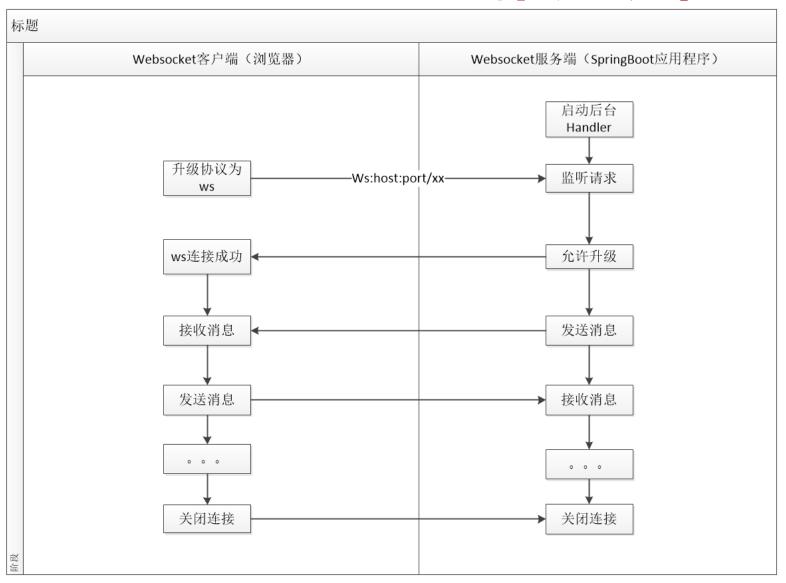
- ◆熟练运用HTML5 WebSocket开发技术,能够在一般实际项目中使用;
- ◆完整掌握SockJS、Stomp、RabbitMQ消息代理技术,可应用到较大型项目中;
- ◆掌握Nginx、SpringCloudGateway多实例负载均衡开发技术,能搭建消息推送多实例集群运行环境。

什么是WebSocket消息推送

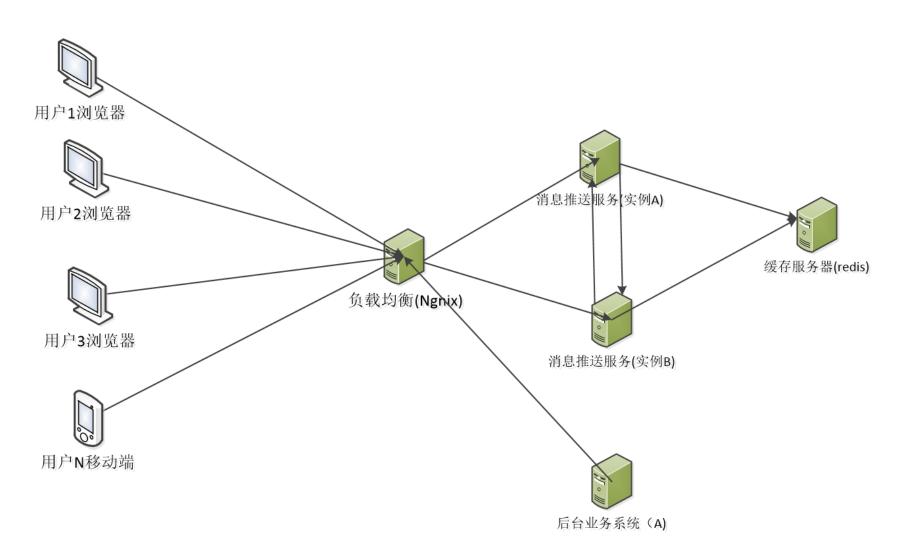
WebSocket 是 HTML5 开始提供的一种在单个 TCP 连接上进行全双工通讯的协议,使得客户端和服务器之间的数据交换变得更加简单,允许服务端主动向客户端推送数据。



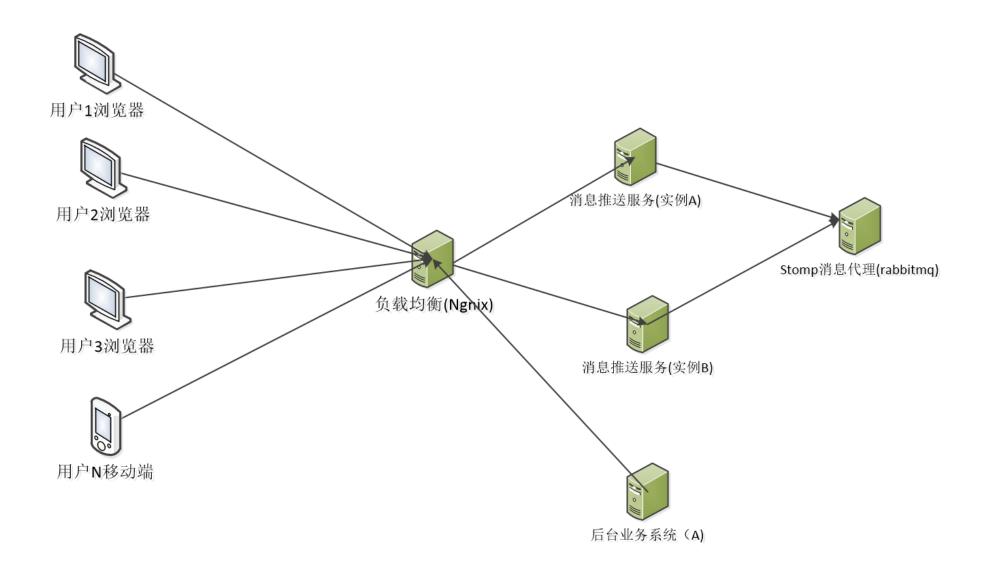
WebSocket消息推送流程



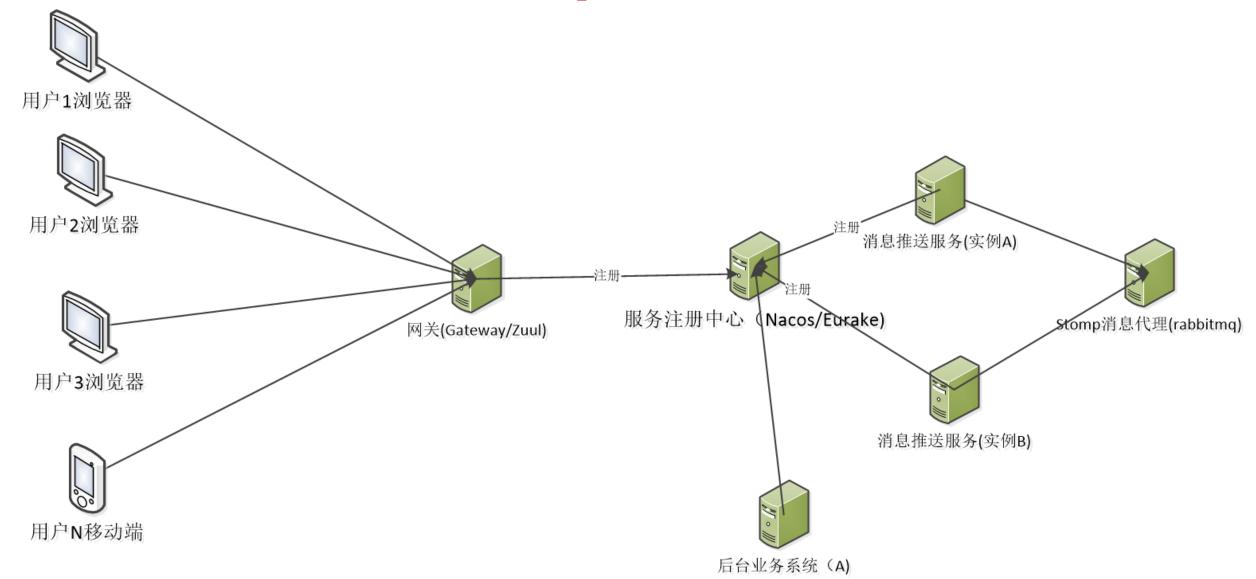
负载均衡(非Stomp方案)



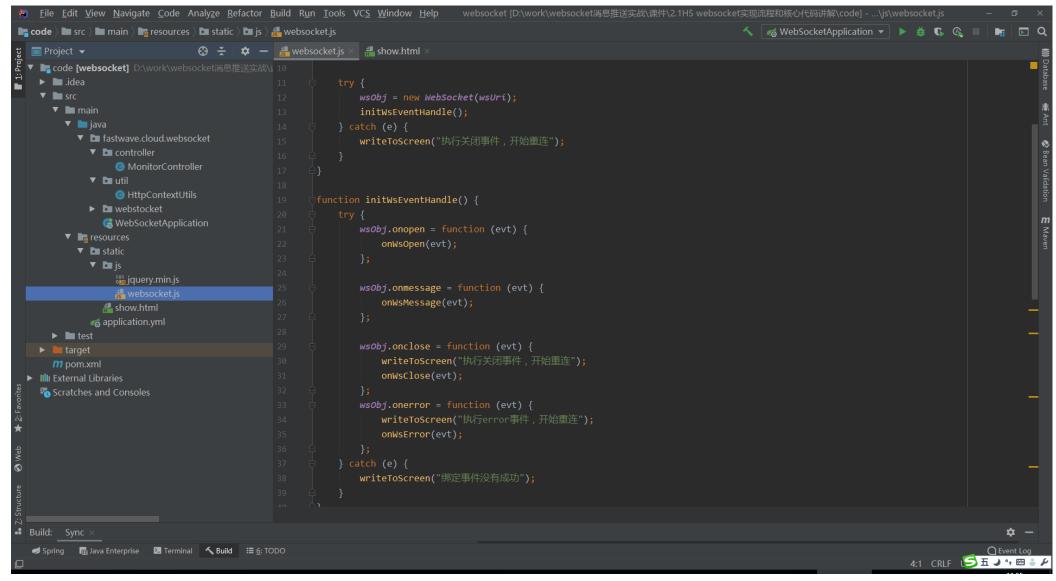
负载均衡(Stomp代理+Nginx方案)



负载均衡(stomp代理+微服务方案)



项目截图(H5原生WebSocket入门)



项目截图(Stomp)

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
websocket [D:\workspaces\fastwave-cloud-stomp] - ...\src\main\resources\static\js\websocket.js - IntelliJ IDEA
Fractional Proof. Fraction 1 In the second of the sec
                                                                                                                                                                                                                                                                                                                                                                 🚜 websocket.js × 🚜 show.html × 💿 WebsocketController.java
     ■ Project ▼
         fastwave-cloud-stomp [websocket] D:\works; 4
         ▶ ■ .idea
          ▼ ■ src
                ▼ I main
                                                                                                                                              userId = GetQueryString( name: "userId");
                      ▼ iava
                            ▼ I fastwave.cloud.stomp
                                                                                                                                              stompClient = Stomp.over(socket);
                                  WebsocketController
                                                                                                                                                        writeToScreen("connected: " + frame);
                                  webstocket
                                                                                                                                                        stompClient.subscribe('/topic/chat', function (response) {
                                         G WebSocketApplication
                                                                                                                                                                   writeToScreen(response.body);
                      ▼ I resources
                            ▼ □ static
                                  ▼ Lis
                                                                                                                                                        stompClient.subscribe("/queue/user " + userId, function (response) {
                                                iguery.min.js
                                                                                                                                                                   writeToScreen(response.body);
                                               # websocket.js
                                         # show.html
                                   application.yml
               ▶ ■ test
                                                                                                                                                                   wsCreateHandler && clearTimeout(wsCreateHandler);
    ► III External Libraries
                                                                                                                                                                  wsCreateHandler = setTimeout( handler: function () {
           Scratches and Consoles
                                                                                                                                                                              console.log("重连...");
                                                                                                                                                                             console.log("重连完成");
                                                                                                                                  function disconnect()

Spring  

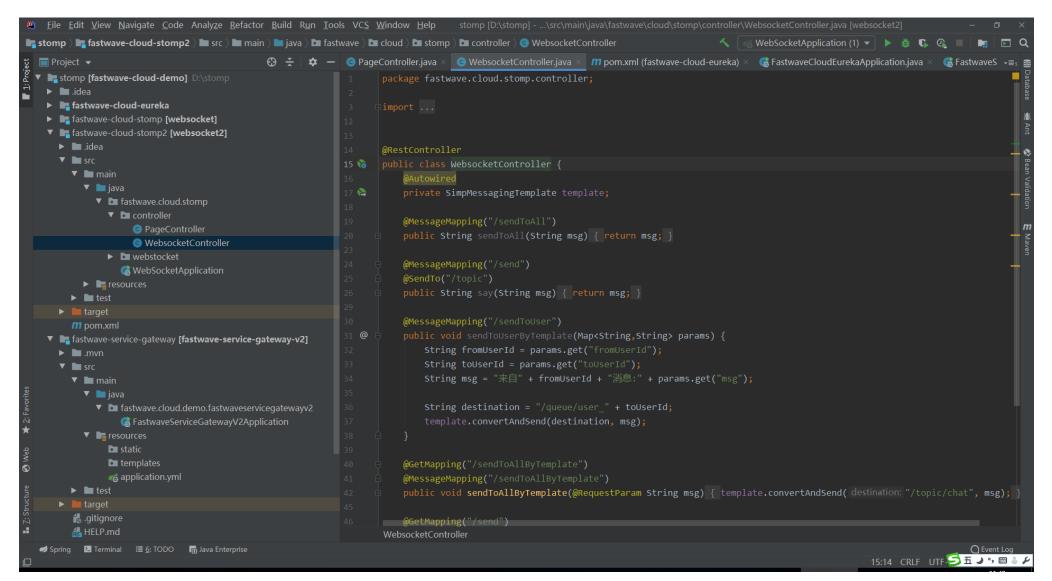
☐ Java Enterprise  

Terminal  

☐ TODO

                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      C Event Log
                                                                                                                                                                                                                                                                                                                                                                                                                                                 4:1 CRLF (与五) 🤊 🖽 🕹 🗲
```

项目截图(微服务)



关联课程

如果您对分布式微服务感兴趣,可以试看本人另一门课程,也有很多免费的课时,名称为:2020SpringCloud微服务架构综合剖析,可以通过本人关联课程或在平台搜索"SpringCloud微服务架构",课程以当前新技术、主流技术为核心。

如果您对分布式消息队列感兴趣,可以试看一下,大家看一看免费的课时就可以入门使用了,

课程名: RabbitMQ消息队列整合SpringBoot分布式事务100%可靠消息发送。