1、大话websocket及课程介绍

简介：

websocket介绍、使用场景分享、学习课程需要什么基础

笔记:

websocket介绍：

WebSocket协议是基于TCP的一种新的网络协议。它实现了浏览器与服务器全双工(full-duplex)通信——允许服务器主动发送信息给客户端

websocket使用场景分享

如弹幕，网页聊天系统，实时监控，股票行情推送等

学习课程需要什么基础:

javaweb基础, html, js, http协议

2、课程技术选型和浏览器兼容讲解

简介：

简单介绍什么是springboot、socketjs、stompjs，及解决使用浏览器兼容问题

笔记：

知识点:

技术框架基本介绍

springboot：

是什么：

1、简化新Spring应用的初始搭建以及开发过程

2、嵌入的Tomcat，无需部署WAR文件

3、简化Maven配置, 自动配置Spring

学习资料：

1、官网 https://projects.spring.io/spring-boot

2、springboot整合websocket资料: https://spring.io/guides/gs/messaging-stomp-websocket/

socketjs：

是什么：

1、是一个浏览器JavaScript库，提供了一个类似WebSocket的对象。

2、提供了一个连贯的跨浏览器的JavaScriptAPI，在浏览器和Web服务器之间创建了一个低延迟，全双工，跨域的通信通道

3、在底层SockJS首先尝试使用本地WebSocket。如果失败了，它可以使用各种浏览器特定的传输协议，并通过类似WebSocket的抽象方式呈现它们

4、SockJS旨在适用于所有现代浏览器和不支持WebSocket协议的环境。

学习资料：

1、git地址：https://github.com/sockjs/sockjs-client

stompjs：

是什么：

1、STOMP Simple (or Streaming) Text Orientated Messaging Protocol

它定义了可互操作的连线格式，以便任何可用的STOMP客户端都可以与任何STOMP消息代理进行通信，以在语言和平台之间提供简单而广泛的消息互操作性（归纳一句话：是一个简单的面向文本的消息传递协议。）

学习资料:

https://stomp-js.github.io/stomp-websocket/codo/class/Client.html#connect-dynamic

3、websocket广播、单播、组播介绍和使用场景说明

简介：

主要讲解websocket的一些概念，如广播，单播等，他们的基本区别和使用场景

笔记：

单播(Unicast):

点对点，私信私聊

广播(Broadcast)(所有人):

游戏公告，发布订阅

多播，也叫组播(Multicast)（特地人群）:

多人聊天，发布订阅

4、广播技术应用 简单websocket游戏公告系统《一》

简介：

快速用springboot搭建一个web应用和maven相关依赖的引入

笔记:

springboot框架搭建和maven依赖

资料地址:https://spring.io/guides/gs/messaging-stomp-websocket/

5、游戏公告系统里maven依赖包webjars使用介绍

简介：介绍什么是webjars，为什么使用这个maven依赖包，解决我们的什么问题

笔记：

1、方便统一管理

2、主要解决前端框架版本不一致，文件混乱等问题

3、把前端资源，打包成jar包，借助maven工具进行管理

6、广播技术应用 简单websocket游戏公告系统《二》

简介：

springBoot里面websocket的配置类讲解，接口开发，页面模板套用

笔记：

页面访问路径：localhost:8080/v1/index.html

7、websocket游戏公告系统流程详解

简介：

公告系统里面的相关socketjs和stompjs API介绍和流程讲解

8、websocket推送两种推送方法的区别和讲解

简介：

讲解websocket推送方式：@SendTo注解和SimpMessagingTemplate的区别

笔记：

1、SendTo 不通用，固定发送给指定的订阅者

2、SimpMessagingTemplate 灵活，支持多种发送方式

9、Springboot针对websocket 4类的监听器介绍和使用

简介：

SpringBoot里面websocekt监听器的使用，包含订阅、取消订阅，socekt连接和断开连接4类监听器的编写和使用

笔记：

注意点：

1、需要监听器类需要实现接口ApplicationListener<T> T表示事件类型，下列几种都是对应的websocket事件类型

2、在监听器类上注解 @Component，spring会把改类纳入管理

websocket模块监听器类型：

SessionSubscribeEvent 订阅事件

SessionUnsubscribeEvent 取消订阅事件

SessionDisconnectEvent 断开连接事件

SessionDisconnectEvent 建立连接事件

10、点对点聊天和简单消息头介绍

简介：

使用socketjs订阅API,进行点对点聊天；StompHeaderAccessor简单介绍

笔记：

11、websocket结合springboot定时推送，实时监控JVM系统负载

简介：

websocket结合springboot的注解Scheduled实现定时推送，使用服务端定时推送注意事项；

开发简单监控JVM监控功能

笔记：

1、在controller的类方法上标注 @Scheduled(fixedRate = 3000) 表示这个方法会定时执行

fixedRate表示是多少毫秒 3000就3秒

2、需要在springboot启动类上@EnableScheduling

3、被注解@Scheduled标记的方法，是不能有参数，不然会报错

12、项目应用之股票行情推送实战《一》 效果展示，阿里云API介绍和httpClient工具类使用

简介：

展示简单股票行情推送的效果，及介绍阿里云API市场，httpClient工具类的使用

笔记:

访问地址：localhost:8080/v5/index.html

阿里云API市场: https://market.aliyun.com/data?spm=5176.8142029.388261.183.346bc16fAs3slP

HttpClientUtils下载: https://github.com/aliyun/api-gateway-demo-sign-java

13、项目应用之股票行情推送实战《二》

简介：相关接口开发，页面展示对接

笔记：

推送接口开发和页面展示

纠错: 只进行推送的controller不用加

“@MessageMapping”注解，@MessageMapping是用于客户端发送数据到服务端的路由配置

14、websocket结合spring相关拦截器使用 拦截器介绍，及HandshakeInterceptor握手拦截器介绍

简介：

拦截器介绍，springBoot结合websocket相关拦截器使用，握手拦截器的开发和使用

笔记：

1、编写一个类，实现一个接口HandshakeInterceptor；写完之后需要在websocket配置里面启用

.addInterceptors(new HttpHandShakeIntecepter())

2、实现两个方法beforeHandshake和afterHandshake，在里面可以获取resuest和response

15、综合项目实战多人聊天系统

简介：

结合前面的课程知识点，开发个综合项目实战，包含的功能模块：用户登录，实时在线用户推送，群聊等

16、用户在线状态维护功能介绍和登录API开发

简介：

用户状态功能相关接口开发和登录API接口开发

17、spring channel拦截器介绍及用户上线下线功能开发

简介：

channel频道拦截器使用讲解，结合StompHeaderAccessor实现用户上线下线功能

笔记:

1、ChannelInterceptorAdapter 频道拦截器适配器,具体实现的接口是ChannelIntecepter

2、需要ChannelInterceptorAdapter子类重写override对应的方法，实现自己的逻辑，主要是

public void postSend(Message<?> message, MessageChannel channel, boolean sent)

3、ChannelInterceptorAdapter子类需要在配置Websocket的配置里面加入

4、在配置类里面加入

@Override

public void configureClientInboundChannel(ChannelRegistration registration) {

registration.interceptors( new SocketChannelIntecepter());

}

@Override

public void configureClientOutboundChannel(ChannelRegistration registration) {

registration.interceptors( new SocketChannelIntecepter());

}

资料：

https://docs.spring.io/spring/docs/current/javadoc-api/org/springframework/messaging/support/ChannelInterceptorAdapter.html

18、实时推送在线用户功能开发

简介：使用schdule注解，推送在线用户接口开发

19、多人群聊功能开发

简介：群聊topic订阅，页面调试功能开发

20、nginx反向代理websocket配置

简介：

nginx配置websocket协议和多节点集群，相关问题和解决方案

笔记：

1、为什么用nginx，支持高并发，解决多节点部署，充当负载均衡使用

2、用 ip\_hash 进行分发到固定后端服务器

3、nginx配置文件里面需要增加下面的配置还有对应的map映射

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection $connection\_upgrade;

21、课程总结和答疑

知乎上相关参考知识：https://www.zhihu.com/question/20215561