target/

!.mvn/wrapper/maven-wrapper.jar

### STS ###

.apt\_generated

.classpath

.factorypath

.project

.settings

.springBeans

### IntelliJ IDEA ###

.idea

\*.iws

\*.iml

\*.ipr

### NetBeans ###

nbproject/private/

build/

nbbuild/

dist/

nbdist/

.nb-gradle/

# Elastic Beanstalk Files

.elasticbeanstalk/\*

!.elasticbeanstalk/\*.cfg.yml

!.elasticbeanstalk/\*.global.yml

distributionUrl=https://repo1.maven.org/maven2/org/apache/maven/apache-maven/3.5.0/apache-maven-3.5.0-bin.zip

# Start with a base image containing Java runtime

FROM openjdk:8-jdk-alpine

# Add Maintainer Info

MAINTAINER Rajeev Kumar Singh <callicoder@gmail.com>

# Add a volume pointing to /tmp

VOLUME /tmp

# Make port 8080 available to the world outside this container

EXPOSE 8080

# The application's jar file

ARG JAR\_FILE=target/websocket-demo-0.0.1-SNAPSHOT.jar

# Add the application's jar to the container

ADD ${JAR\_FILE} websocket-demo.jar

# Run the jar file

ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/websocket-demo.jar"]

## Spring Boot WebSocket Chat Appplication

You can checkout the live version of the application at https://spring-ws-chat.herokuapp.com/

![App Screenshot](screenshot.png)

## Requirements

1. Java - 1.8.x

2. Maven - 3.x.x

## Steps to Setup

\*\*1. Clone the application\*\*

```bash

git clone https://github.com/callicoder/spring-boot-websocket-chat-demo.git

```

\*\*2. Build and run the app using maven\*\*

```bash

cd spring-boot-websocket-chat-demo

mvn package

java -jar target/websocket-demo-0.0.1-SNAPSHOT.jar

```

Alternatively, you can run the app directly without packaging it like so -

```bash

mvn spring-boot:run

```

## Learn More

You can find the tutorial for this application on my blog -

https://www.callicoder.com/spring-boot-websocket-chat-example/

apiVersion: apps/v1 # API version

kind: Deployment # Type of kubernetes resource

metadata:

name: chat-server # Name of the kubernetes resource

labels: # Labels that will be applied to this resource

app: chat-server

spec:

replicas: 1 # No. of replicas/pods to run in this deployment

selector:

matchLabels: # The deployment applies to any pods mayching the specified labels

app: chat-server

template: # Template for creating the pods in this deployment

metadata:

labels: # Labels that will be applied to each Pod in this deployment

app: chat-server

spec: # Spec for the containers that will be run in the Pods

containers:

- name: chat-server

image: callicoder/spring-boot-websocket-chat-demo:0.0.1-SNAPSHOT

imagePullPolicy: IfNotPresent

ports:

- name: http

containerPort: 8080 # The port that the container exposes

---

apiVersion: v1 # API version

kind: Service # Type of the kubernetes resource

metadata:

name: chat-server # Name of the kubernetes resource

labels: # Labels that will be applied to this resource

app: chat-server

spec:

type: NodePort # The service will be exposed by opening a Port on each node and proxying it.

selector:

app: chat-server # The service exposes Pods with label `app=polling-app-server`

ports: # Forward incoming connections on port 8080 to the target port 8080

- name: http

port: 8080

targetPort: 8080

#!/bin/sh

# ----------------------------------------------------------------------------

# Licensed to the Apache Software Foundation (ASF) under one

# or more contributor license agreements. See the NOTICE file

# distributed with this work for additional information

# regarding copyright ownership. The ASF licenses this file

# to you under the Apache License, Version 2.0 (the

# "License"); you may not use this file except in compliance

# with the License. You may obtain a copy of the License at

#

# http://www.apache.org/licenses/LICENSE-2.0

#

# Unless required by applicable law or agreed to in writing,

# software distributed under the License is distributed on an

# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY

# KIND, either express or implied. See the License for the

# specific language governing permissions and limitations

# under the License.

# ----------------------------------------------------------------------------

# ----------------------------------------------------------------------------

# Maven2 Start Up Batch script

#

# Required ENV vars:

# ------------------

# JAVA\_HOME - location of a JDK home dir

#

# Optional ENV vars

# -----------------

# M2\_HOME - location of maven2's installed home dir

# MAVEN\_OPTS - parameters passed to the Java VM when running Maven

# e.g. to debug Maven itself, use

# set MAVEN\_OPTS=-Xdebug -Xrunjdwp:transport=dt\_socket,server=y,suspend=y,address=8000

# MAVEN\_SKIP\_RC - flag to disable loading of mavenrc files

# ----------------------------------------------------------------------------

if [ -z "$MAVEN\_SKIP\_RC" ] ; then

if [ -f /etc/mavenrc ] ; then

. /etc/mavenrc

fi

if [ -f "$HOME/.mavenrc" ] ; then

. "$HOME/.mavenrc"

fi

fi

# OS specific support. $var \_must\_ be set to either true or false.

cygwin=false;

darwin=false;

mingw=false

case "`uname`" in

CYGWIN\*) cygwin=true ;;

MINGW\*) mingw=true;;

Darwin\*) darwin=true

# Use /usr/libexec/java\_home if available, otherwise fall back to /Library/Java/Home

# See https://developer.apple.com/library/mac/qa/qa1170/\_index.html

if [ -z "$JAVA\_HOME" ]; then

if [ -x "/usr/libexec/java\_home" ]; then

export JAVA\_HOME="`/usr/libexec/java\_home`"

else

export JAVA\_HOME="/Library/Java/Home"

fi

fi

;;

esac

if [ -z "$JAVA\_HOME" ] ; then

if [ -r /etc/gentoo-release ] ; then

JAVA\_HOME=`java-config --jre-home`

fi

fi

if [ -z "$M2\_HOME" ] ; then

## resolve links - $0 may be a link to maven's home

PRG="$0"

# need this for relative symlinks

while [ -h "$PRG" ] ; do

ls=`ls -ld "$PRG"`

link=`expr "$ls" : '.\*-> \(.\*\)$'`

if expr "$link" : '/.\*' > /dev/null; then

PRG="$link"

else

PRG="`dirname "$PRG"`/$link"

fi

done

saveddir=`pwd`

M2\_HOME=`dirname "$PRG"`/..

# make it fully qualified

M2\_HOME=`cd "$M2\_HOME" && pwd`

cd "$saveddir"

# echo Using m2 at $M2\_HOME

fi

# For Cygwin, ensure paths are in UNIX format before anything is touched

if $cygwin ; then

[ -n "$M2\_HOME" ] &&

M2\_HOME=`cygpath --unix "$M2\_HOME"`

[ -n "$JAVA\_HOME" ] &&

JAVA\_HOME=`cygpath --unix "$JAVA\_HOME"`

[ -n "$CLASSPATH" ] &&

CLASSPATH=`cygpath --path --unix "$CLASSPATH"`

fi

# For Migwn, ensure paths are in UNIX format before anything is touched

if $mingw ; then

[ -n "$M2\_HOME" ] &&

M2\_HOME="`(cd "$M2\_HOME"; pwd)`"

[ -n "$JAVA\_HOME" ] &&

JAVA\_HOME="`(cd "$JAVA\_HOME"; pwd)`"

# TODO classpath?

fi

if [ -z "$JAVA\_HOME" ]; then

javaExecutable="`which javac`"

if [ -n "$javaExecutable" ] && ! [ "`expr \"$javaExecutable\" : '\([^ ]\*\)'`" = "no" ]; then

# readlink(1) is not available as standard on Solaris 10.

readLink=`which readlink`

if [ ! `expr "$readLink" : '\([^ ]\*\)'` = "no" ]; then

if $darwin ; then

javaHome="`dirname \"$javaExecutable\"`"

javaExecutable="`cd \"$javaHome\" && pwd -P`/javac"

else

javaExecutable="`readlink -f \"$javaExecutable\"`"

fi

javaHome="`dirname \"$javaExecutable\"`"

javaHome=`expr "$javaHome" : '\(.\*\)/bin'`

JAVA\_HOME="$javaHome"

export JAVA\_HOME

fi

fi

fi

if [ -z "$JAVACMD" ] ; then

if [ -n "$JAVA\_HOME" ] ; then

if [ -x "$JAVA\_HOME/jre/sh/java" ] ; then

# IBM's JDK on AIX uses strange locations for the executables

JAVACMD="$JAVA\_HOME/jre/sh/java"

else

JAVACMD="$JAVA\_HOME/bin/java"

fi

else

JAVACMD="`which java`"

fi

fi

if [ ! -x "$JAVACMD" ] ; then

echo "Error: JAVA\_HOME is not defined correctly." >&2

echo " We cannot execute $JAVACMD" >&2

exit 1

fi

if [ -z "$JAVA\_HOME" ] ; then

echo "Warning: JAVA\_HOME environment variable is not set."

fi

CLASSWORLDS\_LAUNCHER=org.codehaus.plexus.classworlds.launcher.Launcher

# traverses directory structure from process work directory to filesystem root

# first directory with .mvn subdirectory is considered project base directory

find\_maven\_basedir() {

if [ -z "$1" ]

then

echo "Path not specified to find\_maven\_basedir"

return 1

fi

basedir="$1"

wdir="$1"

while [ "$wdir" != '/' ] ; do

if [ -d "$wdir"/.mvn ] ; then

basedir=$wdir

break

fi

# workaround for JBEAP-8937 (on Solaris 10/Sparc)

if [ -d "${wdir}" ]; then

wdir=`cd "$wdir/.."; pwd`

fi

# end of workaround

done

echo "${basedir}"

}

# concatenates all lines of a file

concat\_lines() {

if [ -f "$1" ]; then

echo "$(tr -s '\n' ' ' < "$1")"

fi

}

BASE\_DIR=`find\_maven\_basedir "$(pwd)"`

if [ -z "$BASE\_DIR" ]; then

exit 1;

fi

export MAVEN\_PROJECTBASEDIR=${MAVEN\_BASEDIR:-"$BASE\_DIR"}

echo $MAVEN\_PROJECTBASEDIR

MAVEN\_OPTS="$(concat\_lines "$MAVEN\_PROJECTBASEDIR/.mvn/jvm.config") $MAVEN\_OPTS"

# For Cygwin, switch paths to Windows format before running java

if $cygwin; then

[ -n "$M2\_HOME" ] &&

M2\_HOME=`cygpath --path --windows "$M2\_HOME"`

[ -n "$JAVA\_HOME" ] &&

JAVA\_HOME=`cygpath --path --windows "$JAVA\_HOME"`

[ -n "$CLASSPATH" ] &&

CLASSPATH=`cygpath --path --windows "$CLASSPATH"`

[ -n "$MAVEN\_PROJECTBASEDIR" ] &&

MAVEN\_PROJECTBASEDIR=`cygpath --path --windows "$MAVEN\_PROJECTBASEDIR"`

fi

WRAPPER\_LAUNCHER=org.apache.maven.wrapper.MavenWrapperMain

exec "$JAVACMD" \

$MAVEN\_OPTS \

-classpath "$MAVEN\_PROJECTBASEDIR/.mvn/wrapper/maven-wrapper.jar" \

"-Dmaven.home=${M2\_HOME}" "-Dmaven.multiModuleProjectDirectory=${MAVEN\_PROJECTBASEDIR}" \

${WRAPPER\_LAUNCHER} $MAVEN\_CONFIG "$@"

@REM ----------------------------------------------------------------------------

@REM Licensed to the Apache Software Foundation (ASF) under one

@REM or more contributor license agreements. See the NOTICE file

@REM distributed with this work for additional information

@REM regarding copyright ownership. The ASF licenses this file

@REM to you under the Apache License, Version 2.0 (the

@REM "License"); you may not use this file except in compliance

@REM with the License. You may obtain a copy of the License at

@REM

@REM http://www.apache.org/licenses/LICENSE-2.0

@REM

@REM Unless required by applicable law or agreed to in writing,

@REM software distributed under the License is distributed on an

@REM "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY

@REM KIND, either express or implied. See the License for the

@REM specific language governing permissions and limitations

@REM under the License.

@REM ----------------------------------------------------------------------------

@REM ----------------------------------------------------------------------------

@REM Maven2 Start Up Batch script

@REM

@REM Required ENV vars:

@REM JAVA\_HOME - location of a JDK home dir

@REM

@REM Optional ENV vars

@REM M2\_HOME - location of maven2's installed home dir

@REM MAVEN\_BATCH\_ECHO - set to 'on' to enable the echoing of the batch commands

@REM MAVEN\_BATCH\_PAUSE - set to 'on' to wait for a key stroke before ending

@REM MAVEN\_OPTS - parameters passed to the Java VM when running Maven

@REM e.g. to debug Maven itself, use

@REM set MAVEN\_OPTS=-Xdebug -Xrunjdwp:transport=dt\_socket,server=y,suspend=y,address=8000

@REM MAVEN\_SKIP\_RC - flag to disable loading of mavenrc files

@REM ----------------------------------------------------------------------------

@REM Begin all REM lines with '@' in case MAVEN\_BATCH\_ECHO is 'on'

@echo off

@REM enable echoing my setting MAVEN\_BATCH\_ECHO to 'on'

@if "%MAVEN\_BATCH\_ECHO%" == "on" echo %MAVEN\_BATCH\_ECHO%

@REM set %HOME% to equivalent of $HOME

if "%HOME%" == "" (set "HOME=%HOMEDRIVE%%HOMEPATH%")

@REM Execute a user defined script before this one

if not "%MAVEN\_SKIP\_RC%" == "" goto skipRcPre

@REM check for pre script, once with legacy .bat ending and once with .cmd ending

if exist "%HOME%\mavenrc\_pre.bat" call "%HOME%\mavenrc\_pre.bat"

if exist "%HOME%\mavenrc\_pre.cmd" call "%HOME%\mavenrc\_pre.cmd"

:skipRcPre

@setlocal

set ERROR\_CODE=0

@REM To isolate internal variables from possible post scripts, we use another setlocal

@setlocal

@REM ==== START VALIDATION ====

if not "%JAVA\_HOME%" == "" goto OkJHome

echo.

echo Error: JAVA\_HOME not found in your environment. >&2

echo Please set the JAVA\_HOME variable in your environment to match the >&2

echo location of your Java installation. >&2

echo.

goto error

:OkJHome

if exist "%JAVA\_HOME%\bin\java.exe" goto init

echo.

echo Error: JAVA\_HOME is set to an invalid directory. >&2

echo JAVA\_HOME = "%JAVA\_HOME%" >&2

echo Please set the JAVA\_HOME variable in your environment to match the >&2

echo location of your Java installation. >&2

echo.

goto error

@REM ==== END VALIDATION ====

:init

@REM Find the project base dir, i.e. the directory that contains the folder ".mvn".

@REM Fallback to current working directory if not found.

set MAVEN\_PROJECTBASEDIR=%MAVEN\_BASEDIR%

IF NOT "%MAVEN\_PROJECTBASEDIR%"=="" goto endDetectBaseDir

set EXEC\_DIR=%CD%

set WDIR=%EXEC\_DIR%

:findBaseDir

IF EXIST "%WDIR%"\.mvn goto baseDirFound

cd ..

IF "%WDIR%"=="%CD%" goto baseDirNotFound

set WDIR=%CD%

goto findBaseDir

:baseDirFound

set MAVEN\_PROJECTBASEDIR=%WDIR%

cd "%EXEC\_DIR%"

goto endDetectBaseDir

:baseDirNotFound

set MAVEN\_PROJECTBASEDIR=%EXEC\_DIR%

cd "%EXEC\_DIR%"

:endDetectBaseDir

IF NOT EXIST "%MAVEN\_PROJECTBASEDIR%\.mvn\jvm.config" goto endReadAdditionalConfig

@setlocal EnableExtensions EnableDelayedExpansion

for /F "usebackq delims=" %%a in ("%MAVEN\_PROJECTBASEDIR%\.mvn\jvm.config") do set JVM\_CONFIG\_MAVEN\_PROPS=!JVM\_CONFIG\_MAVEN\_PROPS! %%a

@endlocal & set JVM\_CONFIG\_MAVEN\_PROPS=%JVM\_CONFIG\_MAVEN\_PROPS%

:endReadAdditionalConfig

SET MAVEN\_JAVA\_EXE="%JAVA\_HOME%\bin\java.exe"

set WRAPPER\_JAR="%MAVEN\_PROJECTBASEDIR%\.mvn\wrapper\maven-wrapper.jar"

set WRAPPER\_LAUNCHER=org.apache.maven.wrapper.MavenWrapperMain

%MAVEN\_JAVA\_EXE% %JVM\_CONFIG\_MAVEN\_PROPS% %MAVEN\_OPTS% %MAVEN\_DEBUG\_OPTS% -classpath %WRAPPER\_JAR% "-Dmaven.multiModuleProjectDirectory=%MAVEN\_PROJECTBASEDIR%" %WRAPPER\_LAUNCHER% %MAVEN\_CONFIG% %\*

if ERRORLEVEL 1 goto error

goto end

:error

set ERROR\_CODE=1

:end

@endlocal & set ERROR\_CODE=%ERROR\_CODE%

if not "%MAVEN\_SKIP\_RC%" == "" goto skipRcPost

@REM check for post script, once with legacy .bat ending and once with .cmd ending

if exist "%HOME%\mavenrc\_post.bat" call "%HOME%\mavenrc\_post.bat"

if exist "%HOME%\mavenrc\_post.cmd" call "%HOME%\mavenrc\_post.cmd"

:skipRcPost

@REM pause the script if MAVEN\_BATCH\_PAUSE is set to 'on'

if "%MAVEN\_BATCH\_PAUSE%" == "on" pause

if "%MAVEN\_TERMINATE\_CMD%" == "on" exit %ERROR\_CODE%

exit /B %ERROR\_CODE%

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>websocket-demo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>websocket-demo</name>

<description>Spring Boot WebSocket Chat Demo</description>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.2.1.RELEASE</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-websocket</artifactId>

</dependency>

<!-- RabbitMQ Starter Dependency (Not required if you're using the simple in-memory broker for STOMP) -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-amqp</artifactId>

</dependency>

<!-- Following dependency is required for Full Featured STOMP Broker Relay -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-reactor-netty</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

<!-- This plugin is used to create a docker image and publish the image to docker hub-->

<plugin>

<groupId>com.spotify</groupId>

<artifactId>dockerfile-maven-plugin</artifactId>

<version>1.4.0</version>

<configuration>

<!-- replace `callicoder` with your docker id-->

<repository>callicoder/spring-boot-websocket-chat-demo</repository>

<tag>${project.version}</tag>

<buildArgs>

<JAR\_FILE>target/${project.build.finalName}.jar</JAR\_FILE>

</buildArgs>

</configuration>

<executions>

<execution>

<id>default</id>

<phase>install</phase>

<goals>

<goal>build</goal>

<goal>push</goal>

</goals>

</execution>

</executions>

</plugin>

</plugins>

</build>

</project>

package com.example.websocketdemo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class WebsocketDemoApplication {

public static void main(String[] args) {

SpringApplication.run(WebsocketDemoApplication.class, args);

}

}

package com.example.websocketdemo.config;

import org.springframework.context.annotation.Configuration;

import org.springframework.messaging.simp.config.MessageBrokerRegistry;

import org.springframework.web.socket.config.annotation.\*;

/\*\*

\* Created by rajeevkumarsingh on 24/07/17.

\*/

@Configuration

@EnableWebSocketMessageBroker

public class WebSocketConfig implements WebSocketMessageBrokerConfigurer {

@Override

public void registerStompEndpoints(StompEndpointRegistry registry) {

registry.addEndpoint("/ws").withSockJS();

}

@Override

public void configureMessageBroker(MessageBrokerRegistry registry) {

registry.setApplicationDestinationPrefixes("/app");

registry.enableSimpleBroker("/topic"); // Enables a simple in-memory broker

// Use this for enabling a Full featured broker like RabbitMQ

/\*

registry.enableStompBrokerRelay("/topic")

.setRelayHost("localhost")

.setRelayPort(61613)

.setClientLogin("guest")

.setClientPasscode("guest");

\*/

}

}

package com.example.websocketdemo.controller;

import com.example.websocketdemo.model.ChatMessage;

import org.springframework.messaging.handler.annotation.MessageMapping;

import org.springframework.messaging.handler.annotation.Payload;

import org.springframework.messaging.handler.annotation.SendTo;

import org.springframework.messaging.simp.SimpMessageHeaderAccessor;

import org.springframework.stereotype.Controller;

/\*\*

\* Created by rajeevkumarsingh on 24/07/17.

\*/

@Controller

public class ChatController {

@MessageMapping("/chat.sendMessage")

@SendTo("/topic/public")

public ChatMessage sendMessage(@Payload ChatMessage chatMessage) {

return chatMessage;

}

@MessageMapping("/chat.addUser")

@SendTo("/topic/public")

public ChatMessage addUser(@Payload ChatMessage chatMessage,

SimpMessageHeaderAccessor headerAccessor) {

// Add username in web socket session

headerAccessor.getSessionAttributes().put("username", chatMessage.getSender());

return chatMessage;

}

}

package com.example.websocketdemo.controller;

import com.example.websocketdemo.model.ChatMessage;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.context.event.EventListener;

import org.springframework.messaging.simp.SimpMessageSendingOperations;

import org.springframework.messaging.simp.stomp.StompHeaderAccessor;

import org.springframework.stereotype.Component;

import org.springframework.web.socket.messaging.SessionConnectedEvent;

import org.springframework.web.socket.messaging.SessionDisconnectEvent;

/\*\*

\* Created by rajeevkumarsingh on 25/07/17.

\*/

@Component

public class WebSocketEventListener {

private static final Logger logger = LoggerFactory.getLogger(WebSocketEventListener.class);

@Autowired

private SimpMessageSendingOperations messagingTemplate;

@EventListener

public void handleWebSocketConnectListener(SessionConnectedEvent event) {

logger.info("Received a new web socket connection");

}

@EventListener

public void handleWebSocketDisconnectListener(SessionDisconnectEvent event) {

StompHeaderAccessor headerAccessor = StompHeaderAccessor.wrap(event.getMessage());

String username = (String) headerAccessor.getSessionAttributes().get("username");

if(username != null) {

logger.info("User Disconnected : " + username);

ChatMessage chatMessage = new ChatMessage();

chatMessage.setType(ChatMessage.MessageType.LEAVE);

chatMessage.setSender(username);

messagingTemplate.convertAndSend("/topic/public", chatMessage);

}

}

}

package com.example.websocketdemo.model;

/\*\*

\* Created by rajeevkumarsingh on 24/07/17.

\*/

public class ChatMessage {

private MessageType type;

private String content;

private String sender;

public enum MessageType {

CHAT,

JOIN,

LEAVE

}

public MessageType getType() {

return type;

}

public void setType(MessageType type) {

this.type = type;

}

public String getContent() {

return content;

}

public void setContent(String content) {

this.content = content;

}

public String getSender() {

return sender;

}

public void setSender(String sender) {

this.sender = sender;

}

}

\* {

-webkit-box-sizing: border-box;

-moz-box-sizing: border-box;

box-sizing: border-box;

}

html,body {

height: 100%;

overflow: hidden;

}

body {

margin: 0;

padding: 0;

font-weight: 400;

font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;

font-size: 1rem;

line-height: 1.58;

color: #333;

background-color: #f4f4f4;

height: 100%;

}

body:before {

height: 50%;

width: 100%;

position: absolute;

top: 0;

left: 0;

background: #128ff2;

content: "";

z-index: 0;

}

.clearfix:after {

display: block;

content: "";

clear: both;

}

.hidden {

display: none;

}

.form-control {

width: 100%;

min-height: 38px;

font-size: 15px;

border: 1px solid #c8c8c8;

}

.form-group {

margin-bottom: 15px;

}

input {

padding-left: 10px;

outline: none;

}

h1, h2, h3, h4, h5, h6 {

margin-top: 20px;

margin-bottom: 20px;

}

h1 {

font-size: 1.7em;

}

a {

color: #128ff2;

}

button {

box-shadow: none;

border: 1px solid transparent;

font-size: 14px;

outline: none;

line-height: 100%;

white-space: nowrap;

vertical-align: middle;

padding: 0.6rem 1rem;

border-radius: 2px;

transition: all 0.2s ease-in-out;

cursor: pointer;

min-height: 38px;

}

button.default {

background-color: #e8e8e8;

color: #333;

box-shadow: 0 2px 2px 0 rgba(0, 0, 0, 0.12);

}

button.primary {

background-color: #128ff2;

box-shadow: 0 2px 2px 0 rgba(0, 0, 0, 0.12);

color: #fff;

}

button.accent {

background-color: #ff4743;

box-shadow: 0 2px 2px 0 rgba(0, 0, 0, 0.12);

color: #fff;

}

#username-page {

text-align: center;

}

.username-page-container {

background: #fff;

box-shadow: 0 1px 11px rgba(0, 0, 0, 0.27);

border-radius: 2px;

width: 100%;

max-width: 500px;

display: inline-block;

margin-top: 42px;

vertical-align: middle;

position: relative;

padding: 35px 55px 35px;

min-height: 250px;

position: absolute;

top: 50%;

left: 0;

right: 0;

margin: 0 auto;

margin-top: -160px;

}

.username-page-container .username-submit {

margin-top: 10px;

}

#chat-page {

position: relative;

height: 100%;

}

.chat-container {

max-width: 700px;

margin-left: auto;

margin-right: auto;

background-color: #fff;

box-shadow: 0 1px 11px rgba(0, 0, 0, 0.27);

margin-top: 30px;

height: calc(100% - 60px);

max-height: 600px;

position: relative;

}

#chat-page ul {

list-style-type: none;

background-color: #FFF;

margin: 0;

overflow: auto;

overflow-y: scroll;

padding: 0 20px 0px 20px;

height: calc(100% - 150px);

}

#chat-page #messageForm {

padding: 20px;

}

#chat-page ul li {

line-height: 1.5rem;

padding: 10px 20px;

margin: 0;

border-bottom: 1px solid #f4f4f4;

}

#chat-page ul li p {

margin: 0;

}

#chat-page .event-message {

width: 100%;

text-align: center;

clear: both;

}

#chat-page .event-message p {

color: #777;

font-size: 14px;

word-wrap: break-word;

}

#chat-page .chat-message {

padding-left: 68px;

position: relative;

}

#chat-page .chat-message i {

position: absolute;

width: 42px;

height: 42px;

overflow: hidden;

left: 10px;

display: inline-block;

vertical-align: middle;

font-size: 18px;

line-height: 42px;

color: #fff;

text-align: center;

border-radius: 50%;

font-style: normal;

text-transform: uppercase;

}

#chat-page .chat-message span {

color: #333;

font-weight: 600;

}

#chat-page .chat-message p {

color: #43464b;

}

#messageForm .input-group input {

float: left;

width: calc(100% - 85px);

}

#messageForm .input-group button {

float: left;

width: 80px;

height: 38px;

margin-left: 5px;

}

.chat-header {

text-align: center;

padding: 15px;

border-bottom: 1px solid #ececec;

}

.chat-header h2 {

margin: 0;

font-weight: 500;

}

.connecting {

padding-top: 5px;

text-align: center;

color: #777;

position: absolute;

top: 65px;

width: 100%;

}

@media screen and (max-width: 730px) {

.chat-container {

margin-left: 10px;

margin-right: 10px;

margin-top: 10px;

}

}

@media screen and (max-width: 480px) {

.chat-container {

height: calc(100% - 30px);

}

.username-page-container {

width: auto;

margin-left: 15px;

margin-right: 15px;

padding: 25px;

}

#chat-page ul {

height: calc(100% - 120px);

}

#messageForm .input-group button {

width: 65px;

}

#messageForm .input-group input {

width: calc(100% - 70px);

}

.chat-header {

padding: 10px;

}

.connecting {

top: 60px;

}

.chat-header h2 {

font-size: 1.1em;

}

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1.0, minimum-scale=1.0">

<title>Spring Boot WebSocket Chat Application | CalliCoder</title>

<link rel="stylesheet" href="/css/main.css" />

</head>

<body>

<noscript>

<h2>Sorry! Your browser doesn't support Javascript</h2>

</noscript>

<div id="username-page">

<div class="username-page-container">

<h1 class="title">Type your username</h1>

<form id="usernameForm" name="usernameForm">

<div class="form-group">

<input type="text" id="name" placeholder="Username" autocomplete="off" class="form-control" />

</div>

<div class="form-group">

<button type="submit" class="accent username-submit">Start Chatting</button>

</div>

</form>

</div>

</div>

<div id="chat-page" class="hidden">

<div class="chat-container">

<div class="chat-header">

<h2>Spring WebSocket Chat Demo</h2>

</div>

<div class="connecting">

Connecting...

</div>

<ul id="messageArea">

</ul>

<form id="messageForm" name="messageForm" nameForm="messageForm">

<div class="form-group">

<div class="input-group clearfix">

<input type="text" id="message" placeholder="Type a message..." autocomplete="off" class="form-control"/>

<button type="submit" class="primary">Send</button>

</div>

</div>

</form>

</div>

</div>

<script src="https://cdnjs.cloudflare.com/ajax/libs/sockjs-client/1.1.4/sockjs.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/stomp.js/2.3.3/stomp.min.js"></script>

<script src="/js/main.js"></script>

</body>

</html>

'use strict';

var usernamePage = document.querySelector('#username-page');

var chatPage = document.querySelector('#chat-page');

var usernameForm = document.querySelector('#usernameForm');

var messageForm = document.querySelector('#messageForm');

var messageInput = document.querySelector('#message');

var messageArea = document.querySelector('#messageArea');

var connectingElement = document.querySelector('.connecting');

var stompClient = null;

var username = null;

var colors = [

'#2196F3', '#32c787', '#00BCD4', '#ff5652',

'#ffc107', '#ff85af', '#FF9800', '#39bbb0'

];

function connect(event) {

username = document.querySelector('#name').value.trim();

if(username) {

usernamePage.classList.add('hidden');

chatPage.classList.remove('hidden');

var socket = new SockJS('/ws');

stompClient = Stomp.over(socket);

stompClient.connect({}, onConnected, onError);

}

event.preventDefault();

}

function onConnected() {

// Subscribe to the Public Topic

stompClient.subscribe('/topic/public', onMessageReceived);

// Tell your username to the server

stompClient.send("/app/chat.addUser",

{},

JSON.stringify({sender: username, type: 'JOIN'})

)

connectingElement.classList.add('hidden');

}

function onError(error) {

connectingElement.textContent = 'Could not connect to WebSocket server. Please refresh this page to try again!';

connectingElement.style.color = 'red';

}

function sendMessage(event) {

var messageContent = messageInput.value.trim();

if(messageContent && stompClient) {

var chatMessage = {

sender: username,

content: messageInput.value,

type: 'CHAT'

};

stompClient.send("/app/chat.sendMessage", {}, JSON.stringify(chatMessage));

messageInput.value = '';

}

event.preventDefault();

}

function onMessageReceived(payload) {

var message = JSON.parse(payload.body);

var messageElement = document.createElement('li');

if(message.type === 'JOIN') {

messageElement.classList.add('event-message');

message.content = message.sender + ' joined!';

} else if (message.type === 'LEAVE') {

messageElement.classList.add('event-message');

message.content = message.sender + ' left!';

} else {

messageElement.classList.add('chat-message');

var avatarElement = document.createElement('i');

var avatarText = document.createTextNode(message.sender[0]);

avatarElement.appendChild(avatarText);

avatarElement.style['background-color'] = getAvatarColor(message.sender);

messageElement.appendChild(avatarElement);

var usernameElement = document.createElement('span');

var usernameText = document.createTextNode(message.sender);

usernameElement.appendChild(usernameText);

messageElement.appendChild(usernameElement);

}

var textElement = document.createElement('p');

var messageText = document.createTextNode(message.content);

textElement.appendChild(messageText);

messageElement.appendChild(textElement);

messageArea.appendChild(messageElement);

messageArea.scrollTop = messageArea.scrollHeight;

}

function getAvatarColor(messageSender) {

var hash = 0;

for (var i = 0; i < messageSender.length; i++) {

hash = 31 \* hash + messageSender.charCodeAt(i);

}

var index = Math.abs(hash % colors.length);

return colors[index];

}

usernameForm.addEventListener('submit', connect, true)

messageForm.addEventListener('submit', sendMessage, true)

package com.example.websocketdemo;

import org.junit.Test;

import org.junit.runner.RunWith;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.test.context.junit4.SpringRunner;

@RunWith(SpringRunner.class)

@SpringBootTest

public class WebsocketDemoApplicationTests {

@Test

public void contextLoads() {

}

}