예시 CommentPartWithWebSocketV4

헤더에 다음의 스크립트 필요, 메시지 처리 필요

jQuery 필요

<script type="text/javascript">

**var** ws = **new** WebSocket("ws://localhost:9090/websocket/echo.do"); // 전역 변수로 선언

$(document).ready(**function**(){

connectWS();

});

**function** connectWS(){

ws.onmessage = onMessage; //메세지 동작할 때

ws.onclose = onClose;

ws.onerror = onError;

ws.onopen = **function**(event){

console.log("커넥션 생성");

};

}

**function** onMessage(event){

console.log("메세지 작동");

**var** data = event.data;

console.log(data);

// 메세지를 받아서 알람 처리하는 동작 필요!

}

**function** onClose(event){

**if** (event.wasClean) {

console.log('[close] 커넥션이 정상적으로 종료되었습니다(code=${event.code} reason=${event.reason})');

} **else** {

// 예시: 프로세스가 죽거나 네트워크에 장애가 있는 경우

// event.code가 1006이 됩니다.

console.log('[close] 커넥션이 죽었습니다.');

}

}

**function** onError(event){

console.log("Server Error");

}

</script>

1. 댓글을 다는 버튼에 이 동작을 수행하도록 설정
2. 버튼을 누르면 3가지 정보가 메세징되도록 설정

예시) mem\_num,brd\_num,true

mem\_num을 통해 알림을 받아야 하는 사람이 누구인지 전송

brd\_num을 통해 알림을 받은 맴버가 알림을 클릭할 시 대상 글로 이동하도록(추후)

true는 그냥 알림이 있음을 나타내는 것, 삭제하고 mem\_num,brd\_num해도 될 듯

<script type="text/javascript">

$(document).ready(**function**(){

$("#alert-web").on("click", **function**(evt){

evt.preventDefault();

**if**(ws.readyState != 1) **return**;

let msg = $("#test\_mem\_num").val()+","+$("#test\_brd\_num").val()+",true";

ws.send(msg); //소켓 메세징

});

});

</script>

<div>

<input type="button" value="alert" id="alert-web"/>

<p>mem\_num</p>

<input type="text" id="test\_mem\_num" />

<p>brd\_num</p>

<input type="text" id="test\_brd\_num" />

</div>

company.blind.web에 다음의 웹소캣 핸들러 생성

🡺WebSocketHandler.java

package company.blind.web;

import java.util.Collections;

import java.util.HashSet;

import java.util.Set;

import org.springframework.web.socket.TextMessage;

import org.springframework.web.socket.WebSocketSession;

import org.springframework.web.socket.handler.TextWebSocketHandler;

public class ***WebSocketHandler*** extends ***TextWebSocketHandler*** {

// 클라이언트 최초 웹소켓 요청시 생성

private ***Set***<***WebSocketSession***> sessions = ***Collections***.***synchronizedSet***(new ***HashSet***<***WebSocketSession***>());

***@Override***

protected void *handleTextMessage*(***WebSocketSession*** session, ***TextMessage*** message) throws ***Exception*** {

// 메세지 처리부

***System***.***out***.***printf***("%s 로부터 [%s]받음", session.***getId***(), message.***getPayload***());

***System***.***out***.***println***();

session.***sendMessage***(new ***TextMessage***(message.***getPayload***()));

}

***@Override***

public void *afterConnectionEstablished*(***WebSocketSession*** session) throws ***Exception*** {

// 연결 요청 접수 -> 해당 클라이언트와 통신을 담당하는 WebSocketSession 객체 전달

// WebSocketSession 객체를 Set에 담아둠

***System***.***out***.***printf***("%s 연결됨", session.***getId***());

***System***.***out***.***println***();

sessions.***add***(session);

}

***@Override***

public void *handleTransportError*(***WebSocketSession*** session, ***Throwable*** exception) throws ***Exception*** {

// 클라이언트와 데이터 송수신중 에러가 발생시 통신담당 객체를 HashSet에서 제거

sessions.***remove***(session);

}

}

Servlet-context.xml에 다음의 빈 추가

<!-- websocket handler -->

<beans:bean id="echoHandler" class="company.blind.web.WebSocketHandler" />

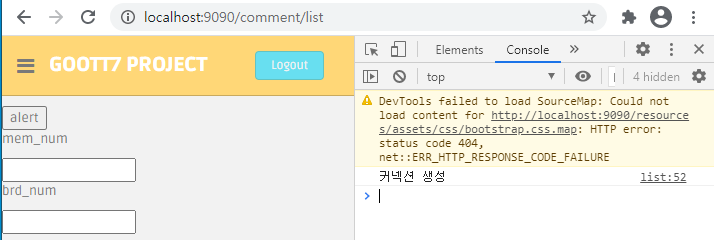
<websocket:handlers>

<websocket:mapping handler="echoHandler" path="/websocket/echo.do" />

</websocket:handlers>

동작

1. 시작, 커넥션이 생성됨



1. 웹소캣 메세징을 하면 다음과 같이 핸들러에서 데이터를 받아옴



1. 동시에 브라우저의 열린 웹 소캣으로 메시지를 전송함, 콘솔로 확인

