

생각에 취하는 날



Alpreah

이론? 그런거 몰라 그냥 따라해!
Mail : Alpreah@gmail.com
Since 2016. 05. 01 ~

!(Spring)/스프링 강의

[Spring] Node.js를 이용해 Spring에 채팅방을 만 자!

2018. 12. 3. 17:38

글을 읽기전에 꼭 읽어주세요

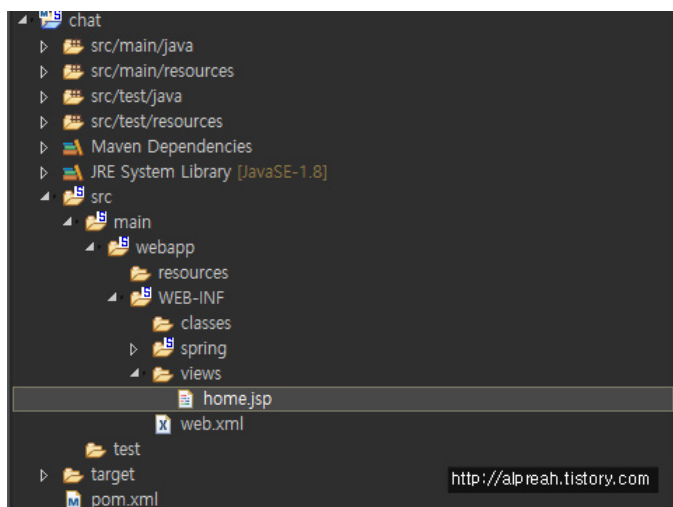
만약, 이전글을 읽고오시지 않았다면 아래의 링크를 눌러 주신뒤 진행을 해주세요.

[Spring] Node.js socket을 통해 Spring서버에 socket을 연결하자!

-- 완성본 파일 --

 소켓 스프링 예제.zip

 소켓 Node.js 예제.zip



우선, STS를 실행 시키신뒤 이전에 만들어둔 home.jsp를 열어주세요.

생각에 취하는 날



Alpreah

이론? 그런거 몰라 그냥 따라해!
Mail : Alpreah@gmail.com
Since 2016. 05. 01 ~

```
%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>

!DOCTYPE html>
html>
head>
meta charset="UTF-8">
title>소켓 서버</title>

style>
.chat_box {
    width: 800px;
    min-width: 800px;
    height: 500px;
    min-height: 500px;
    border: 1px solid black;
}

.msg {
    width: 700px;
}

.msg_process {
    width: 90px;
}

/style>
/head>
body>
<div id="chat_box"></div>
<input type="text" id="msg">
<button id="msg_process">전송</button>

<script src="http://localhost:82/socket.io/socket.io.js"></script>
<script src="https://code.jquery.com/jquery-1.11.1.js"></script>
<script>
$(document).ready(function(){
    var socket = io("http://localhost:82");

    //msg에서 키를 누를때
    $("#msg").keydown(function(key){
        //해당하는 키가 엔터키(13) 일때
        if(key.keyCode == 13){
            //msg_process를 클릭해준다.
            msg_process.click();
        }
    });

    //msg_process를 클릭할 때
    $("#msg_process").click(function(){
        //소켓에 send_msg라는 이벤트로 input에 #msg의 벨류를 담고 보내준다.
        socket.emit("send_msg", $("#msg").val());
        //#msg에 벨류값을 비워준다.
        $("#msg").val("");
    });
});
</script>
/body>
/html>
```

Draw by Color Scripter

cs위

위 소스코드를 home.jsp에 삽입을 해주신뒤 서버를 실행 해주세요.

생각에 취하는 날

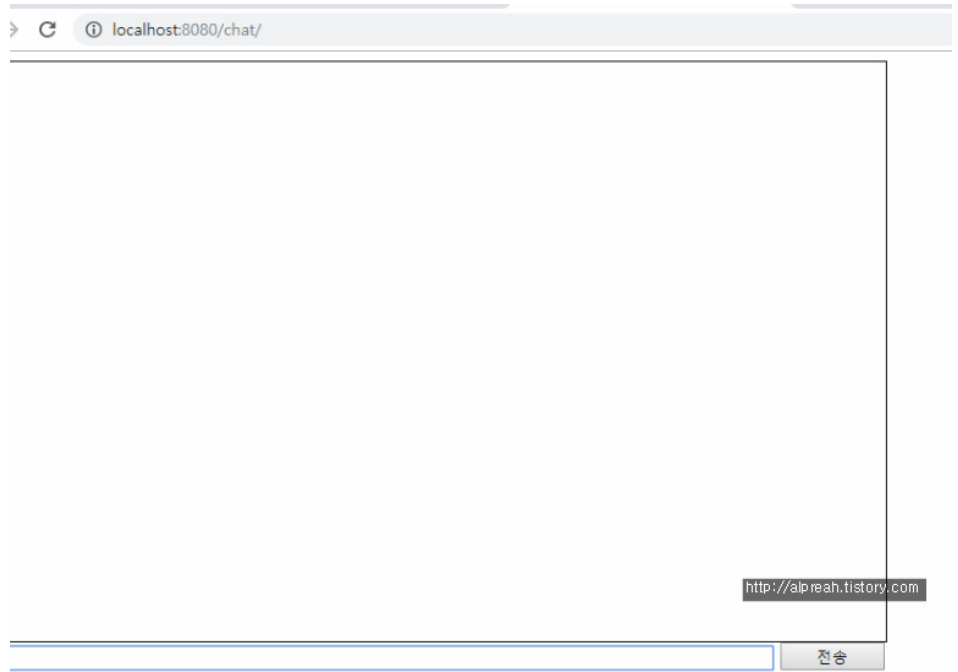


Alpreah

이론? 그런거 몰라 그냥 따라해!

Mail : Alpreah@gmail.com

Since 2016. 05. 01 ~



위 이미지는 실행한 결과입니다.

```
var app = require('express')();
var http = require('http').Server(app);
var io = require('socket.io')(http);

app.get('/', function (req, res) {
  res.send('<h1>안녕하세요 "/" 경로 입니다.</h1>');
});

io.on('connection', function (socket) {
  console.log('한명의 유저가 접속을 했습니다. ');
  socket.on('disconnect', function () {
    console.log('한명의 유저가 접속해제를 했습니다. ');
  });

  socket.on('send_msg', function (msg) {
    console.log(msg);
  });
});

http.listen(82, function () {
  console.log('listening on *:82');
});
```

Colored by Color Scripter CS

위 소스코드는 node.js에 있는 server.js에 넣어주세요.



그런뒤 node.js 서버를 실행 시켜주세요. "node server.js"

그런뒤 아래의 동영상을 참고하셔서 테스트 해주세요.

생각에 취하는 날



Alpreah

이론? 그런거 몰라 그냥 따라해!

Mail : Alpreah@gmail.com

Since 2016. 05. 01 ~

```

<script>
    msg {
        width: 700px;
    }
    msg_process {
        width: 90px;
    }
</script>
</head>
<body>
    <div id="chat_box"></div>
    <input type="text" id="msg">
    <button id="msg_process">전송</button>

    <script src="http://localhost:82/socket.io/socket.io.js"></script>
    <script src="https://code.jquery.com/jquery-1.11.1.js"></script>
    <script>
        $(document).ready(function() {
            var socket = io("http://localhost:82");

            //msg에서 키를 누를때
            $("#msg").keydown(function(key) {
                //해당하는 키가 엔터키(13) 일때
                if (key.keyCode == 13) {
                    //msg_process를 클릭해준다.
                    msg_process.click();
                }
            });

            //msg_process를 클릭할 때
            $("#msg_process").click(function() {
                //소켓에 send_msg라는 이벤트로 input에 #msg의 벨류를 담고 보내준다.
                socket.emit("send_msg", $("#msg").val());
                //#msg에 벨류값을 비워준다.
                $("#msg").val("");
            });

            //소켓 서버로 부터 send_msg를 통해 이벤트를 받을 경우
            socket.on('send_msg', function(msg) {
                //div 태그를 만들어 텍스트를 msg로 지정을 한뒤 #chat_box에 추가를 시켜준다.
                $('<div></div>').text(msg).appendTo("#chat_box");
            });
        });
    </script>
</body>
</html>

```

파란색으로 색칠을 해둔 소스코드를 복사하신뒤 알맞게 넣어주세요.

저장 후 STS서버를 다시 켜주세요.

```

var app = require('express')();
var http = require('http').Server(app);
var io = require('socket.io')(http);

app.get('/', function (req, res) {
    res.send('<h1>안녕하세요 "/" 경로 입니다.</h1>');
});

io.on('connection', function (socket) {
    console.log('한명의 유저가 접속을 했습니다. ');
    socket.on('disconnect', function () {
        console.log('한명의 유저가 접속해제를 했습니다. ');
    });

    socket.on('send_msg', function (msg) {
        //콘솔로 출력을 한다.
        console.log(msg);
        //다시, 소켓을 통해 이벤트를 전송한다.
        io.emit('send_msg', msg);
    });
});

http.listen(82, function () {
    console.log('listening on *:82');
});

```

Colored by Color Scripter cs

이제는 NODE JS로 이동해신뒤 server.js에 파란색으로 칠해둔 소스코드를 넣어주세요.

생각에 취하는 날



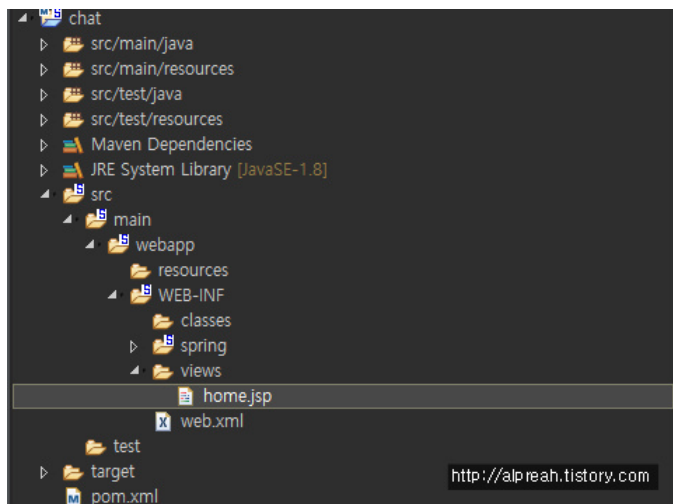
Alpreah

이론? 그런거 몰라 그냥 따라해!
Mail : Alpreah@gmail.com
Since 2016. 05. 01 ~



상과 위에있는 소스코드는 JSP에서 WS(웹소켓)를 연결하고 NODE JS서버로 이벤트를 전송하는 테스트 영상입니다.

이제는 메시지를 div에 출력을 해볼게요



다시, STS로 이동하신뒤 home.jsp를 열어주세요.

```
%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>소켓 서버</title>

<style>
.chat_box {
    width: 800px;
    min-width: 800px;
    height: 500px;
    min-height: 500px;
    border: 1px solid black;
```



생각에 취하는 날




Alpreah

이론? 그런거 몰라 그냥 따라해!
Mail : Alpreah@gmail.com
Since 2016. 05. 01 ~

Comments


의문인사람 2019.05.07 09:35
타 컴퓨터에서 접속해서 사용할려고할때 socket.io를 가져올수없는 에러가 발생하는데 어떻게 처리하면될까요

 Alpreah 2019.05.07 11:36 신고
자바스크립트를 참조 할때
`https://아이피:포트/(예제 경로 복사)`

자바스크립트 구문 중
`var socket = io("http://아이피:포트");`

호랑이가호랑호랑 2019.06.26 14:32
잘보고가용 디스크~

ㅇㅅㅇ 2020.06.12 17:47
톰캣 서버 하나만 돌려도 웹소켓 사용가능한데 굳이 톰캣과 노드js 두가지 서버를 동시에 돌리는 이유가 뭔가요?

 Alpreah 2020.06.13 15:03 신고
저도 자세하게는 잘 모르겠지만, 비교적 가벼운거를 사용할 때는 아파치 웹소켓을 사용하고 조금 무게가 있는것은 노드 JS로 빼는걸로 알고 있어요.

그리고, 글을 쓴 취지는 서버를 따로 두고도 웹소켓을 연결 할 수 있다는을 보여주기 위함 입니다!

여러분의 소중한 댓글을 입력해주세요

Secret Send

```
server: ~$ cd Desktop#Node.js 예제#chat> node server.js
node on *:82 http://alpreah.tistory.com
```

그런뒤 node.js 서버를 실행 시켜주세요. "node server.js"

생각에 취하는 날

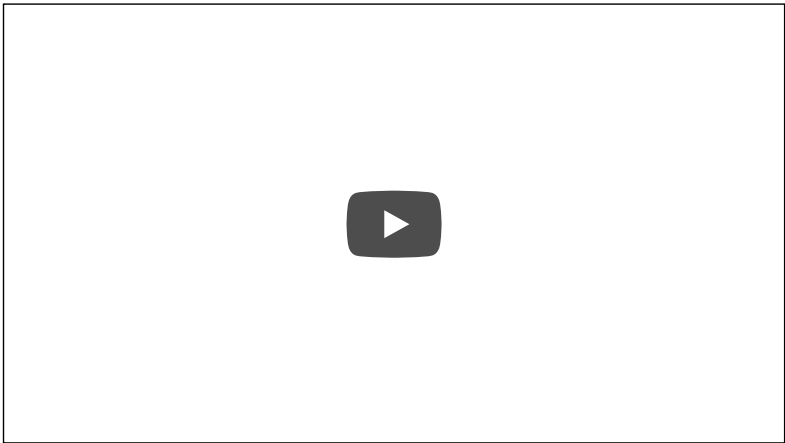


Alpreah

이론? 그런거 몰라 그냥 따라해!
Mail : Alpreah@gmail.com
Since 2016. 05. 01 ~

위와 동일하게 작업을 하셨다면 간단하게 소켓을 사용해서 채팅방을 만들었습니다.

완성본의 결과는 아래의 동영상을 참고해주세요.



5

구독하기

링(Spring) > 스프링 강의' 카테고리의 다른 글		
g) 자동 등록 방식을 구현하자 - google reCAPTCHA -2	(3)	2018.12.09
g) 자동 등록 방식을 구현하자 - google reCAPTCHA -1	(0)	2018.12.07
g) Node.js를 이용해 Spring에 채팅방을 만들자!	(5)	2018.12.03
g) Node.js socket을 통해 Spring서버에 socket을 연결하자!	(4)	2018.12.03
g) 스프링 마이바티스 마리아디비 연결	(2)	2018.11.27
g) Maria DB 설치	(0)	2018.11.27

sp 채팅방, node.js 채팅방, nodejs, socket 채팅방, Socket.io, spring socket coconnect, spring 웹소켓, SPRING 채팅방, STS 채팅방, 소켓사용법, 스프링 소켓 연결, 스프링 소켓통신, 스프링 채팅방, 채팅방 만들기

링(Spring)/스프링 강의' Related Articles

- [Spring] 자동 등록 방식을 구현하자 - google reCAPTCHA -2
- [Spring] 자동 등록 방식을 구현하자 - google reCAPTCHA -1
- [Spring] Node.js socket을 통해 Spring서버에 socket을 연결하자!
- [Spring] 스프링 마이바티스 마리아디비 연결