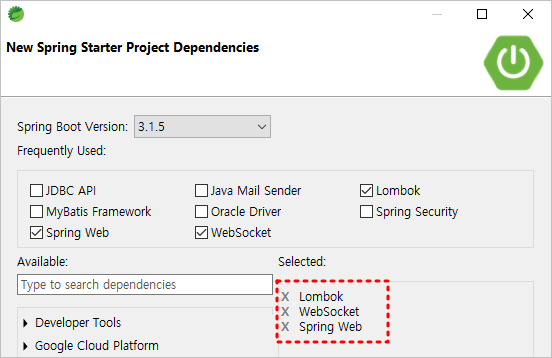
Websocket을 이용한 멀티채팅

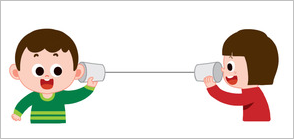
# 프로젝트명 : B14WebSocket

## 준비사항

* 의존설정 : Spring Web, Websocket, Lombok
* JSP 사용을 위한 설정을 한다.
* Refresh Gradle Project 를 눌러 적용한다.

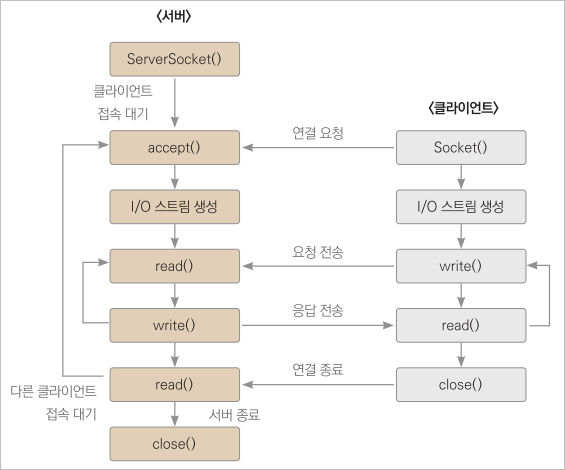


## 소켓이란..??

* 소켓(Socket)은 네트워크에서 동작하는 프로그램의 종착점(endpoint)이라고 주로 표현
* IP 주소와 포트 번호로 이루어져 있으며, 서버와 클라이언트가 양방향 통신을 할 수 있게 해주는 소프트웨어 장치

어린시절 가지고 놀던 종이컵 전화기에서 실은 네트워크의 역할을 하고, 종이컵이 바로 소켓의 역할을 담당한다.

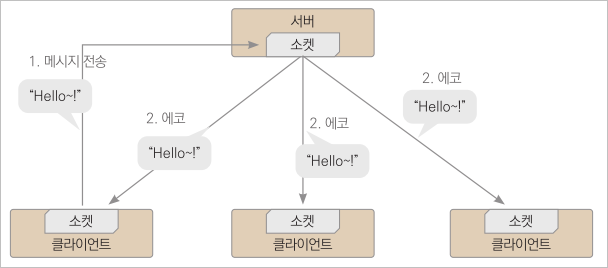
## 자바에서의 소켓 통신 절차



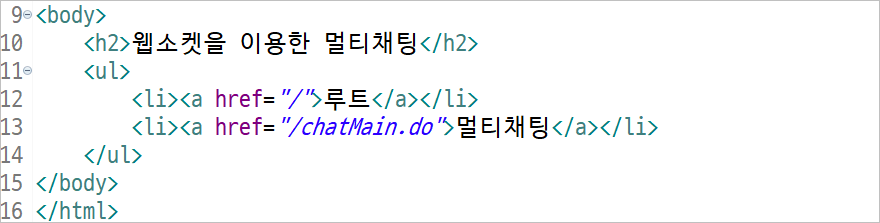
## 웹소켓이란..??

* 일반적인 웹 환경은 클라이언트의 요청을 받으면 응답 후 바로 연결을 종료하는 비연결(connectionless) 동기 소켓 방식을 사용
* 하지만 웹소켓(WebSocket)은 클라이언트의 요청에 응답한 후에도 연결을 그대로 유지하는 연결 지 향(connection oriented) 방식을 사용
* 따라서 별도의 요청이 없어도 서버는 원하면 언제든 클라이언트로 데이터를 전송할 수 있다.

## 에코 클라이언트/서버 모델



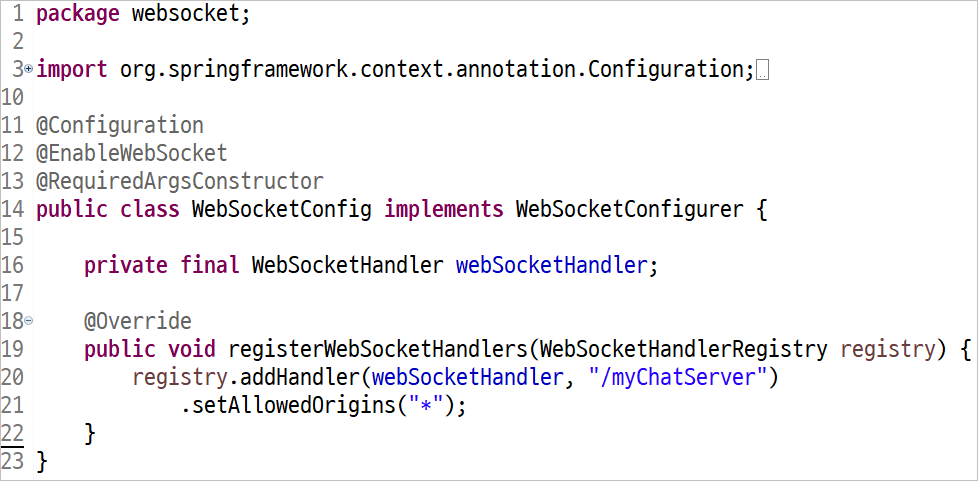
### 뷰 : webapp/WEB-INF/views/home.jsp



### 컨트롤러 : com.edu.springboot.MainController.java

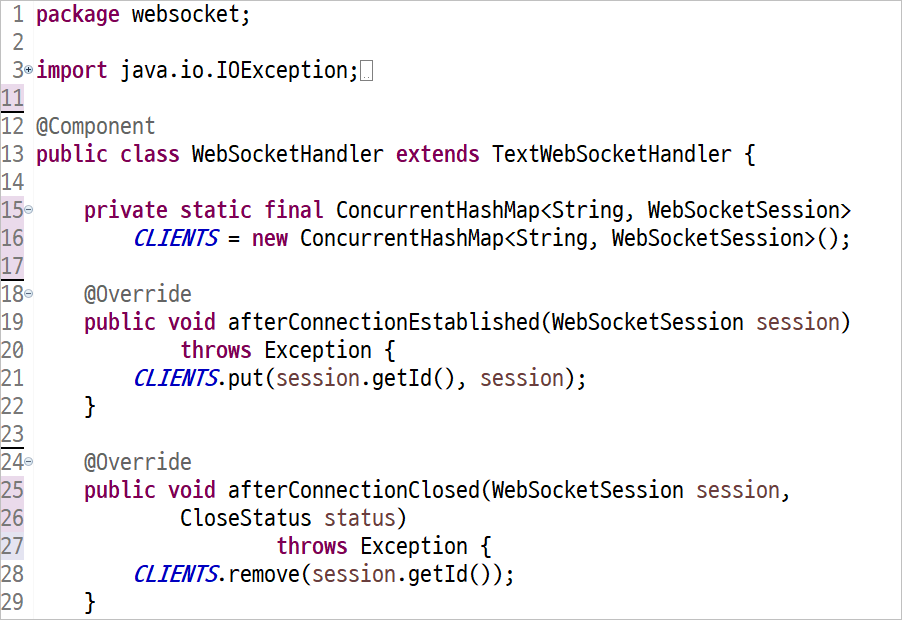


### 소켓설정 : com.edu.springboot.websocket.WebSocketConfig.java



**여기까지 작성하세요.**

### 소켓핸들러 : com.edu.springboot.websocket.WebSocketHandler.java





### 뷰 : /B14WebSocket/src/main/webapp/WEB-INF/views/chatMain.jsp



### 뷰 : /B14WebSocket/src/main/webapp/WEB-INF/views/chatWindow.jsp (포트번호확인)



