



# LINE CHAT (WEB)

신한DS SW 아카데미  
**김신영**

2024. 11. 25

# \* Contents

1. 프로젝트 배경
2. 개발 환경 및 사용 기술
3. 시스템 구조
4. ERD
5. 기능 소개

# \* 프로젝트 목적



기존 CLI 채팅 프로그램 에서  
web server로 변경



WebSocket  
학습 및 활용

# \* 개발 환경 및 사용 기술

## Language

- Java
- Javascript
- CSS
- HTML

## Library

- Lombok

## Tool

- IntelliJ
- Oracle SQL Developer

## Database

- Oracle 11g

## Java API

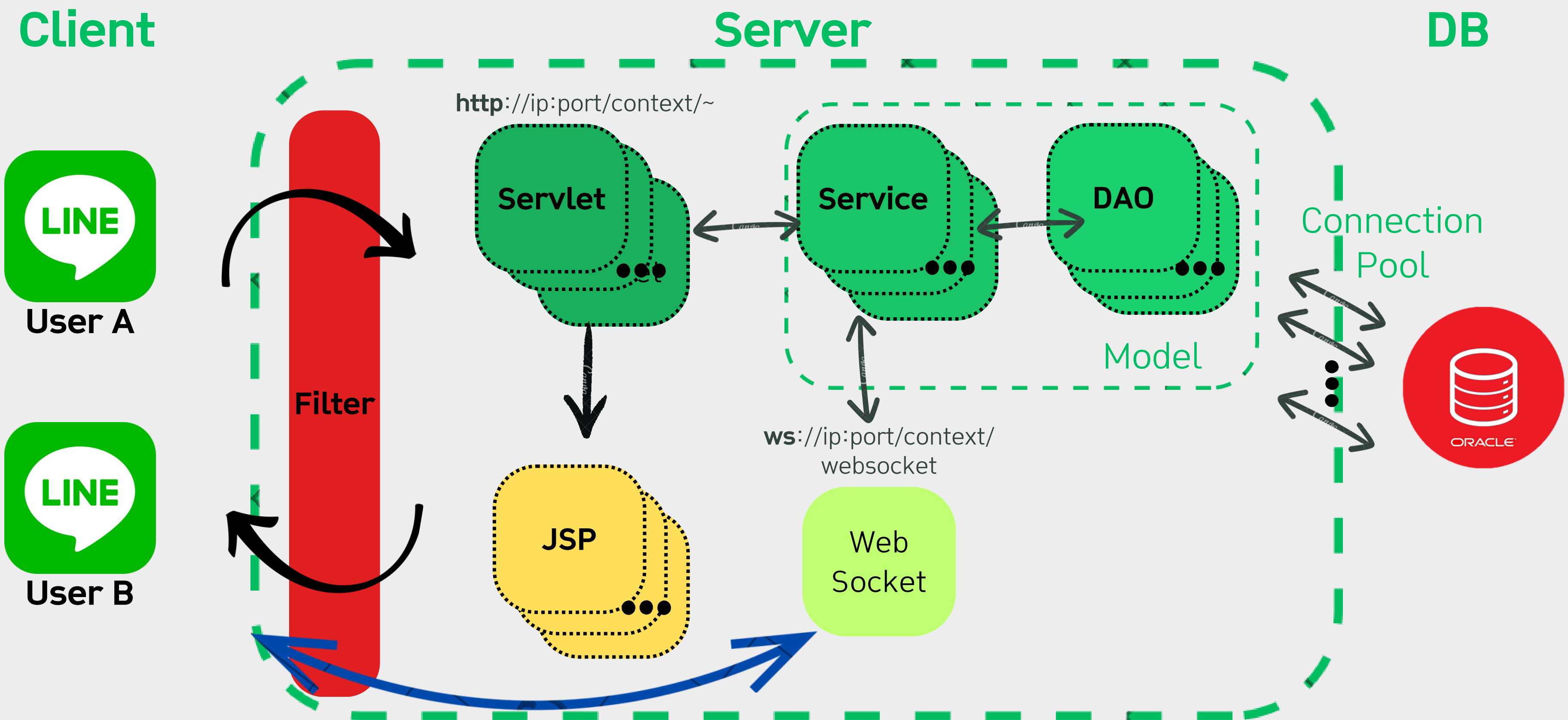
- JDBC
- **WebSocket**

## WAS

- Tomcat 9.0



# \* 시스템 구조



# \* 시스템 구조 - WEB SOCKET

## 1. Web Socket 생성 & 연결

### Client

```
// WebSocket 오브젝트 생성 (자동으로 접속 시작한다. - onopen 함수 호출)  
var websocket = new WebSocket("ws://localhost:9090/LINEchat/websocket?roomId=${roomId}");
```

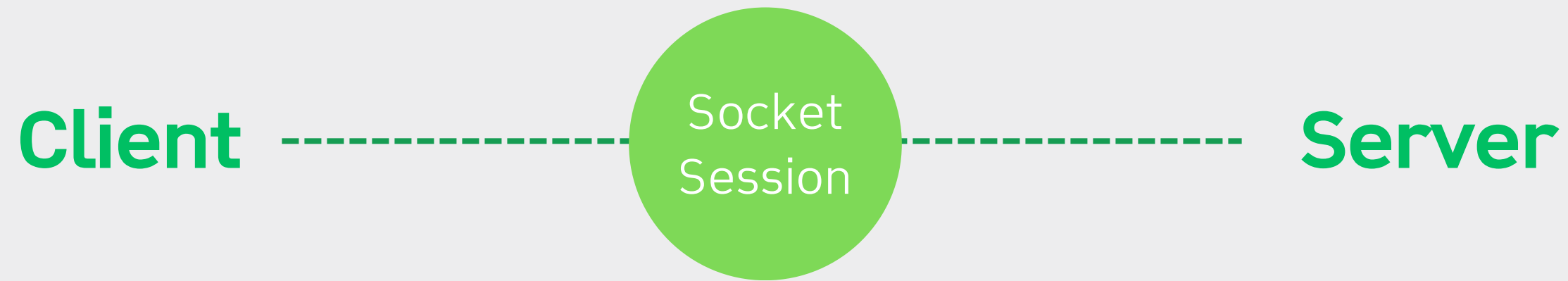
### Server

```
@OnOpen  
public void handleOpen(Session session, EndpointConfig config) {
```

## 2. Session 관리

```
@ServerEndpoint(value = "/websocket", configurator = HttpSessionConfigurator.class)  
public class WebSocket {  
    // 방별로 세션을 관리하기 위한 ConcurrentHashMap  
    5 usages  
    private static final Map<String, Set<Session>> roomSessions = new ConcurrentHashMap<>();  
    3 usages  
    private static final Map<Session, String> sessionRooms = new ConcurrentHashMap<>();  
    // 특정 사용자 정보 관리 위한 ConcurrentHashMap  
    3 usages  
    private static final Map<Session, MemberDTO> sessionUsers = new ConcurrentHashMap<>();
```

# \* 시스템 구조 - WEB SOCKET



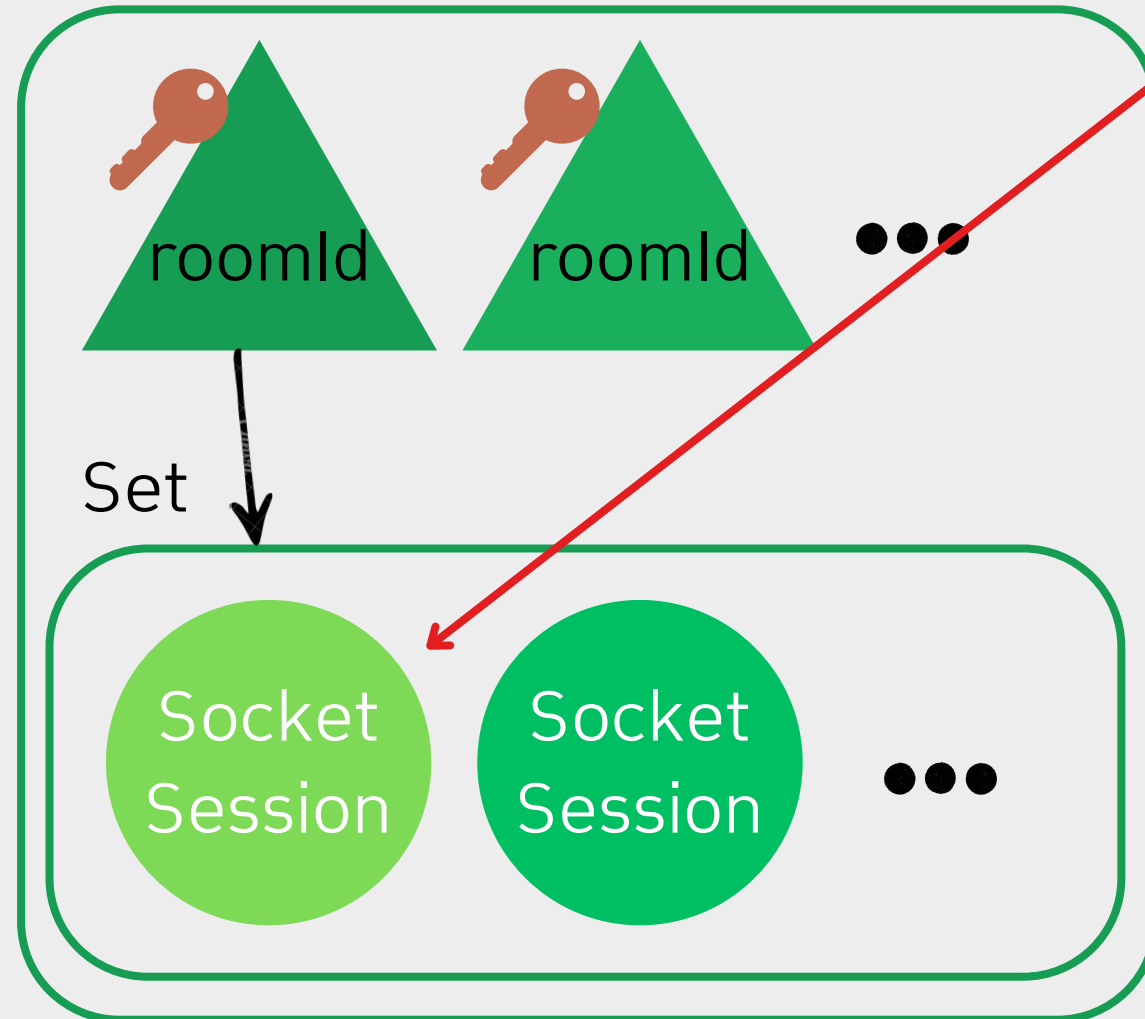
# \* 시스템 구조 - WEB SOCKET

Client

Socket  
Session

Server

roomSession





# \* 시스템 구조 - WEB SOCKET

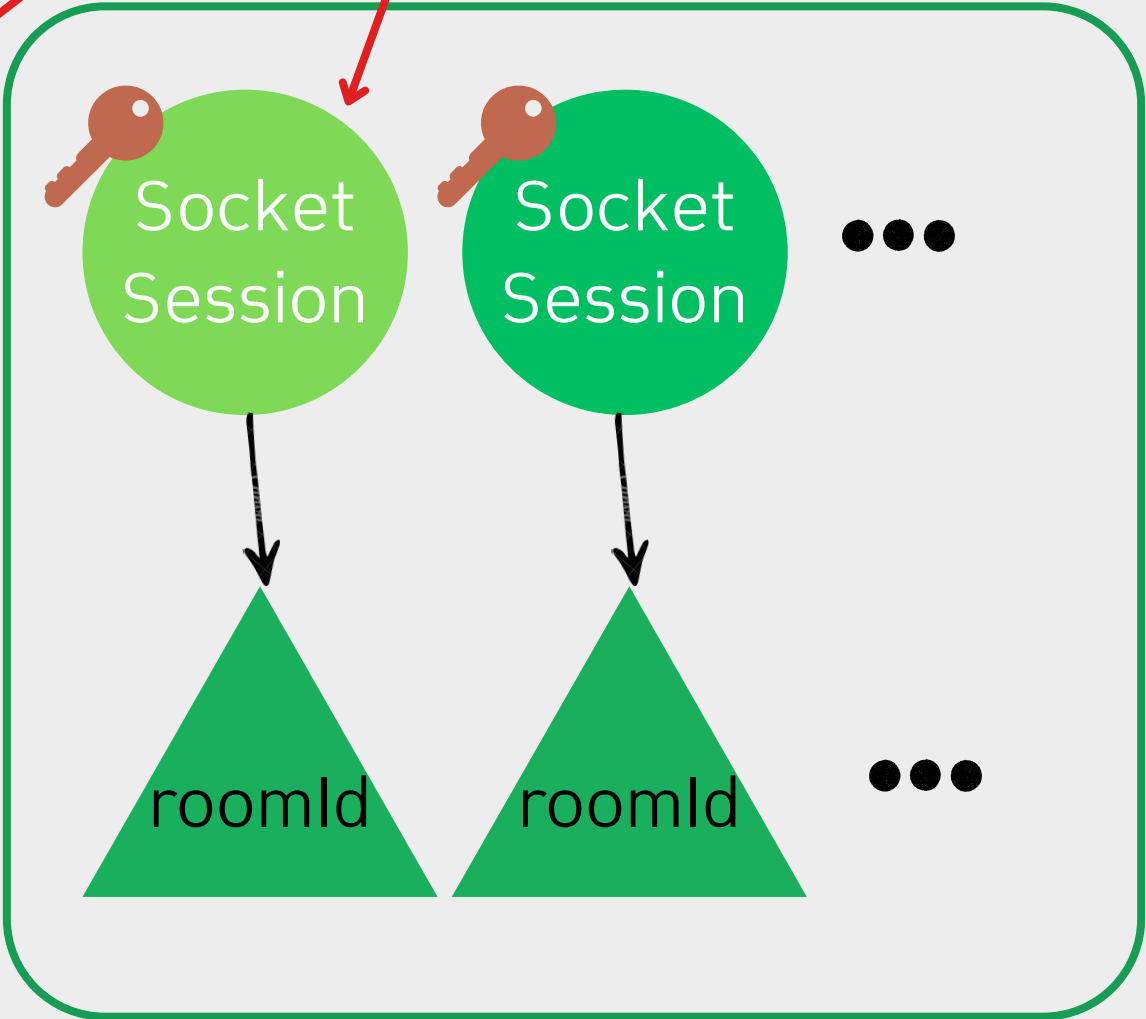
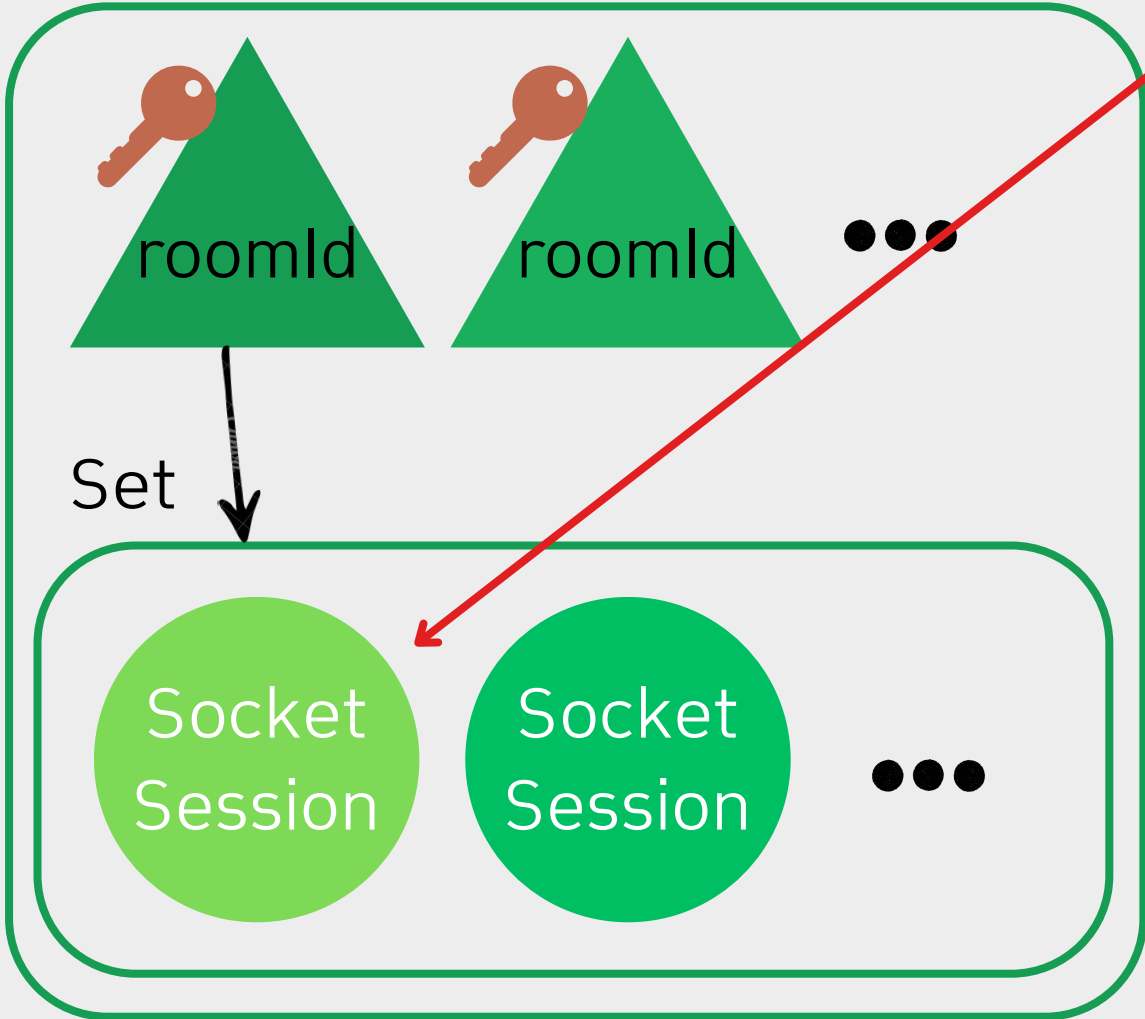
Client

Socket  
Session

Server

roomSession

sessionRoom



# \* 시스템 구조 - WEB SOCKET

Client

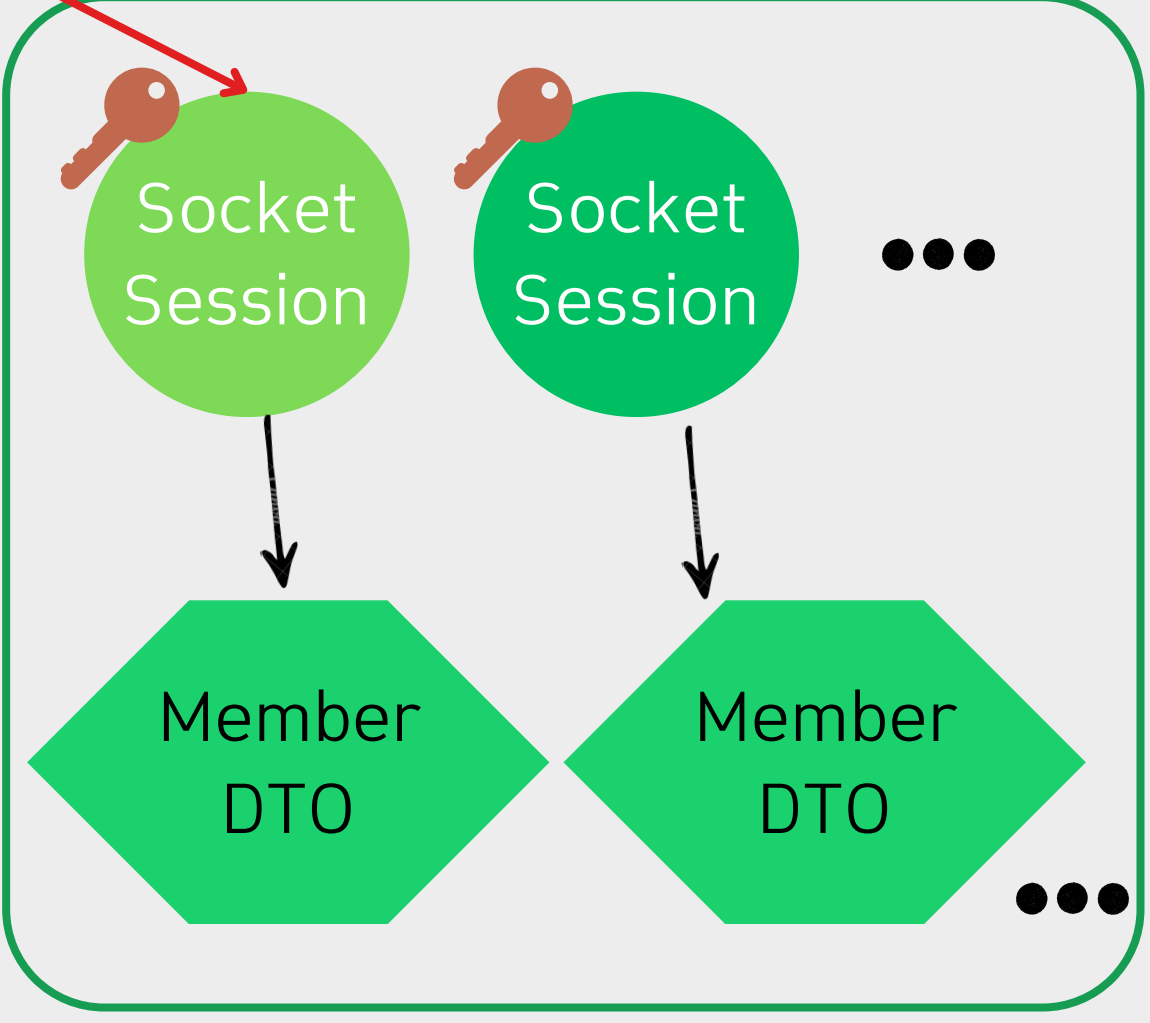
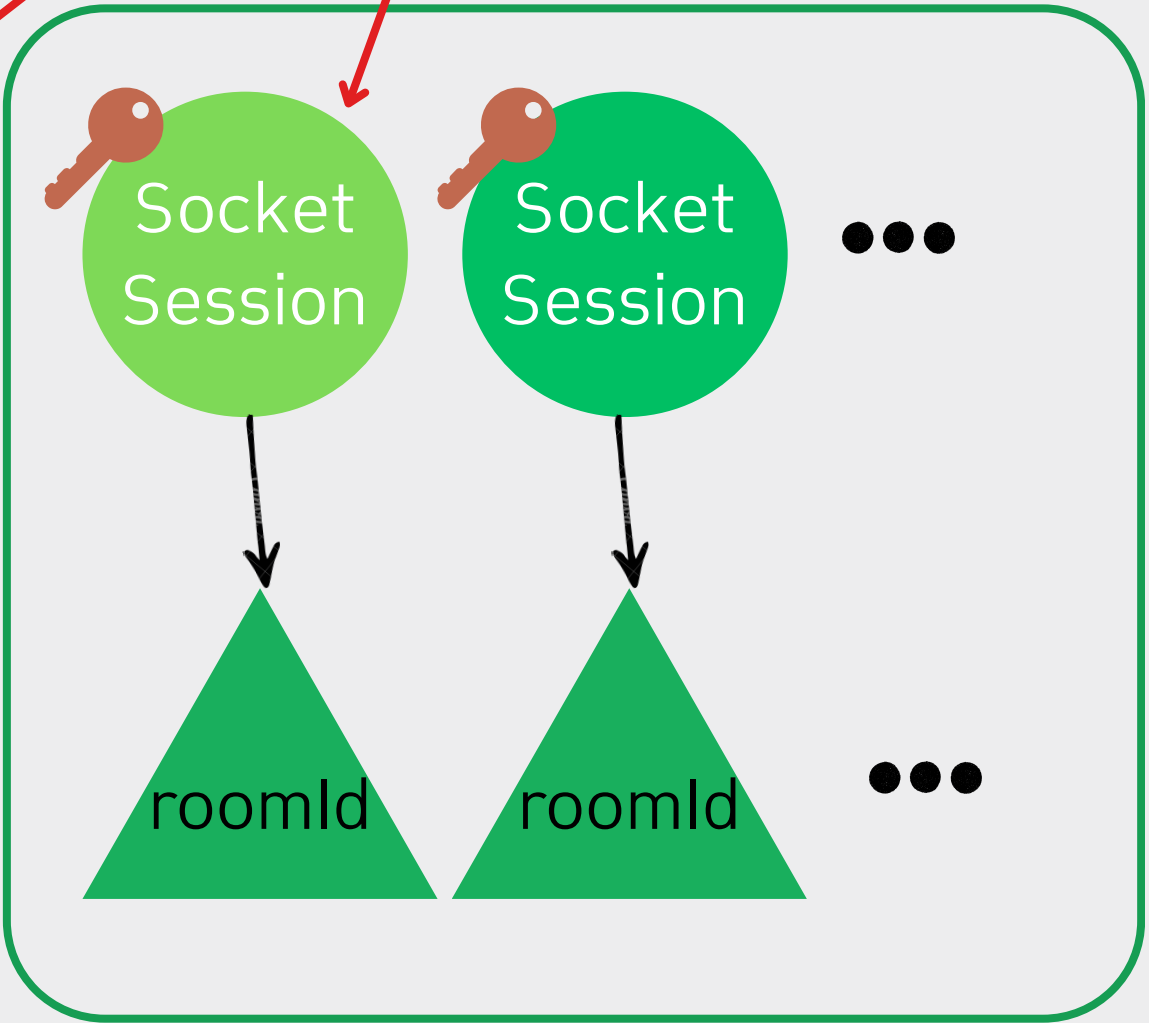
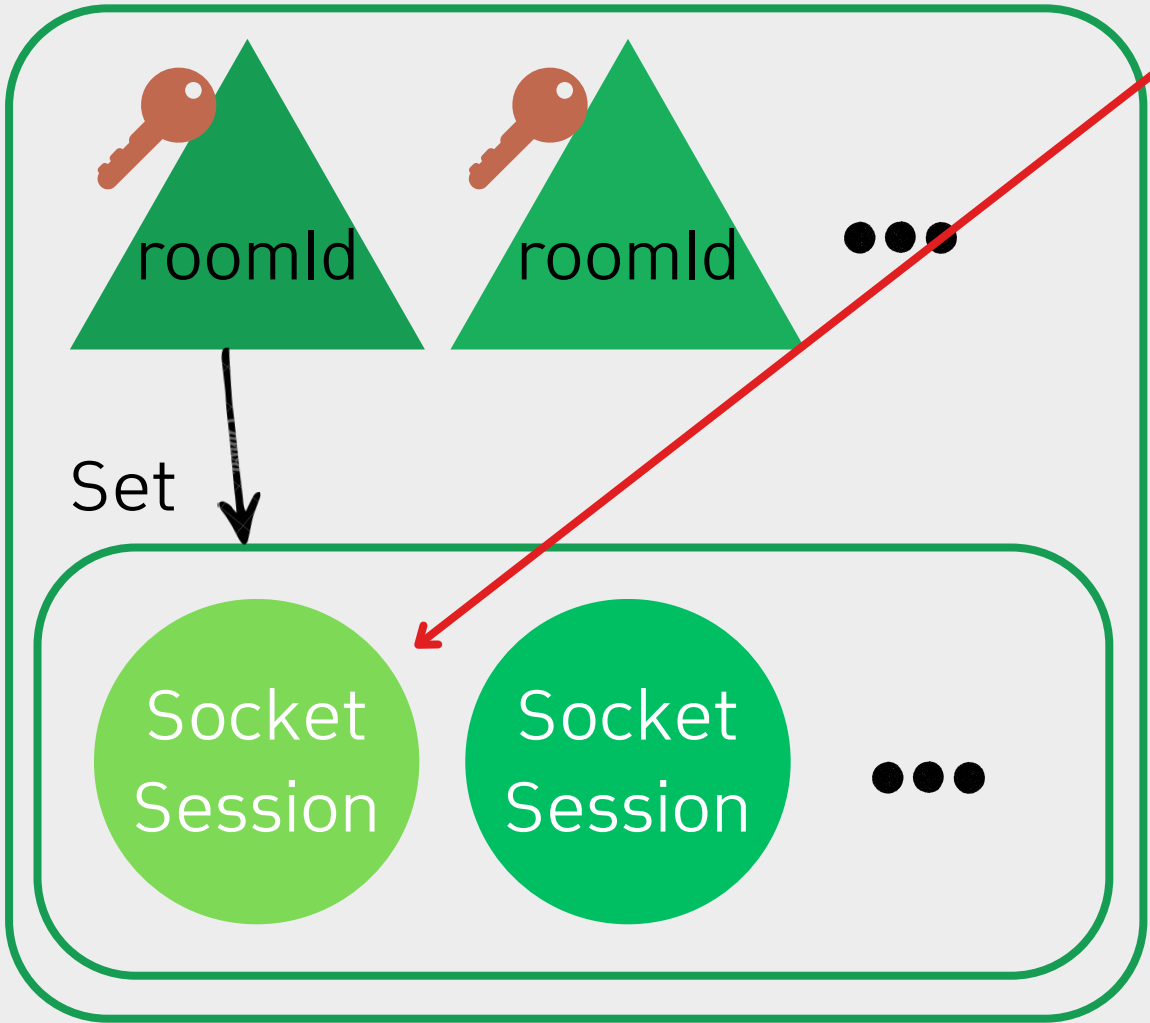
Server



roomSession

sessionRoom

sessionUsers



# \* 시스템 구조 - WEB SOCKET

## @ServerEndpoint

```
@ServerEndpoint(value = "/websocket", configurator = HttpSessionConfigurator.class)  
public class WebSocket {
```

## @OnOpen

```
@OnOpen  
public void handleOpen(Session session, EndpointConfig config) {
```

## @OnClose

```
@OnClose  
public void handleClose(Session session) {
```

## @OnMessage

```
@OnMessage  
public void handleMessage(String message, Session session) {
```

## @OnError

```
@OnError  
public void handleError(Throwable t) {
```



CLICK HERE

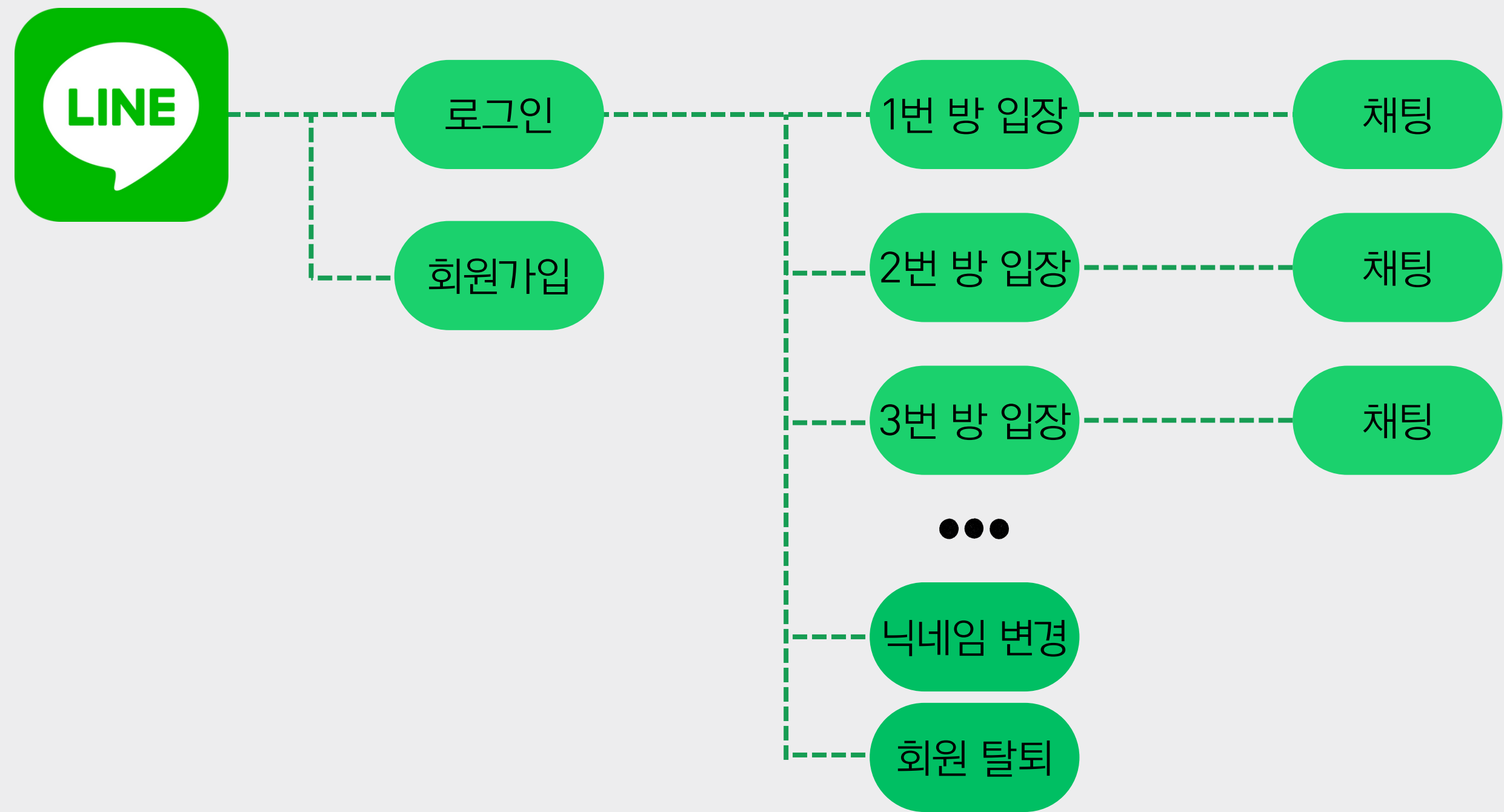


Java Docs  
(javax.websocket)

# \* ERD



# \* 기능 소개



# \* 기능 소개

회원 가입

Line chat x +

localhost:9090/LINEchat/auth/login.do

## 회원가입

아이디

Enter Email

이 입력란을 작성하세요.

비밀번호

Enter Password

닉네임

Repeat Password

Cancel 회원가입

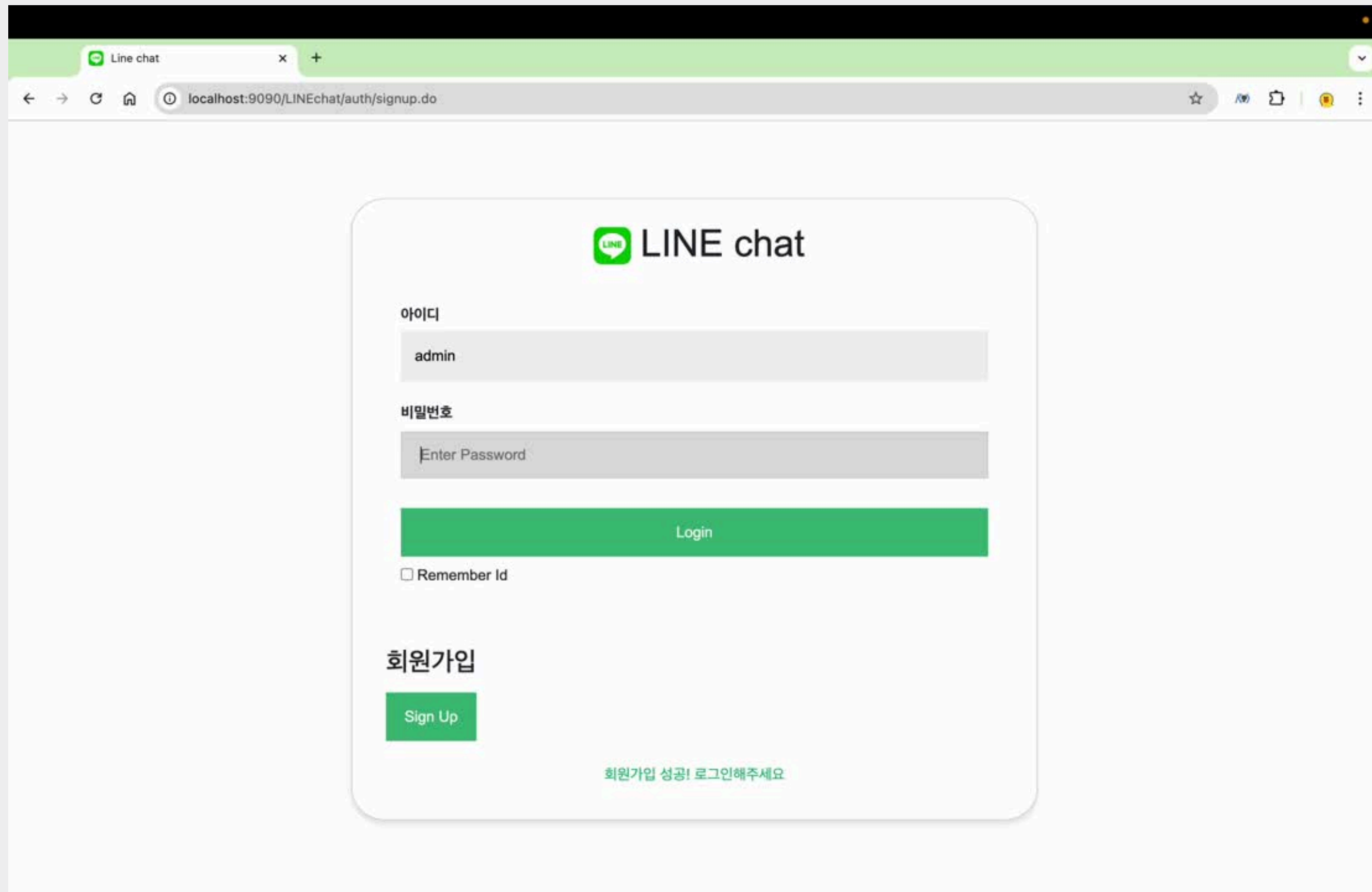
# \* 기능 소개

로그인

#없는 아이디

# 틀린 비밀번호

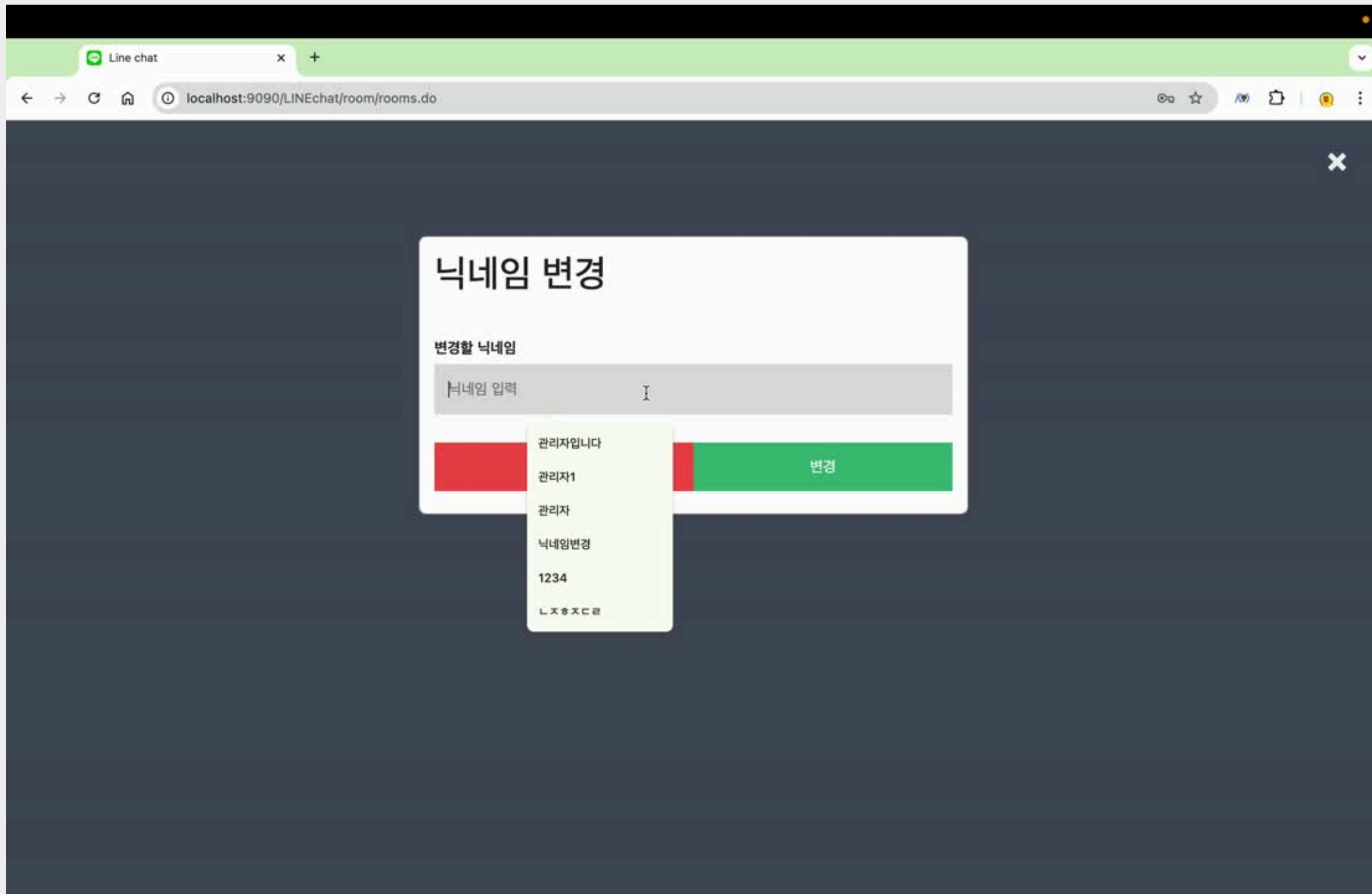
# 아이디 저장



The screenshot shows a web browser window with a single tab titled "Line chat". The address bar displays "localhost:9090/LINEchat/auth/signup.do". The main content area features a white rounded rectangle with the "LINE chat" logo at the top. Below the logo, there are two input fields: "아이디" (ID) with the text "admin" and "비밀번호" (Password) with the placeholder "Enter Password". A green "Login" button is positioned below these fields. Underneath the button is a checkbox labeled "Remember Id". At the bottom of the form, there is a "회원가입" (Sign Up) section with a green "Sign Up" button. A green message at the very bottom of the form reads "회원가입 성공! 로그인해주세요" (Sign up successful! Please log in).

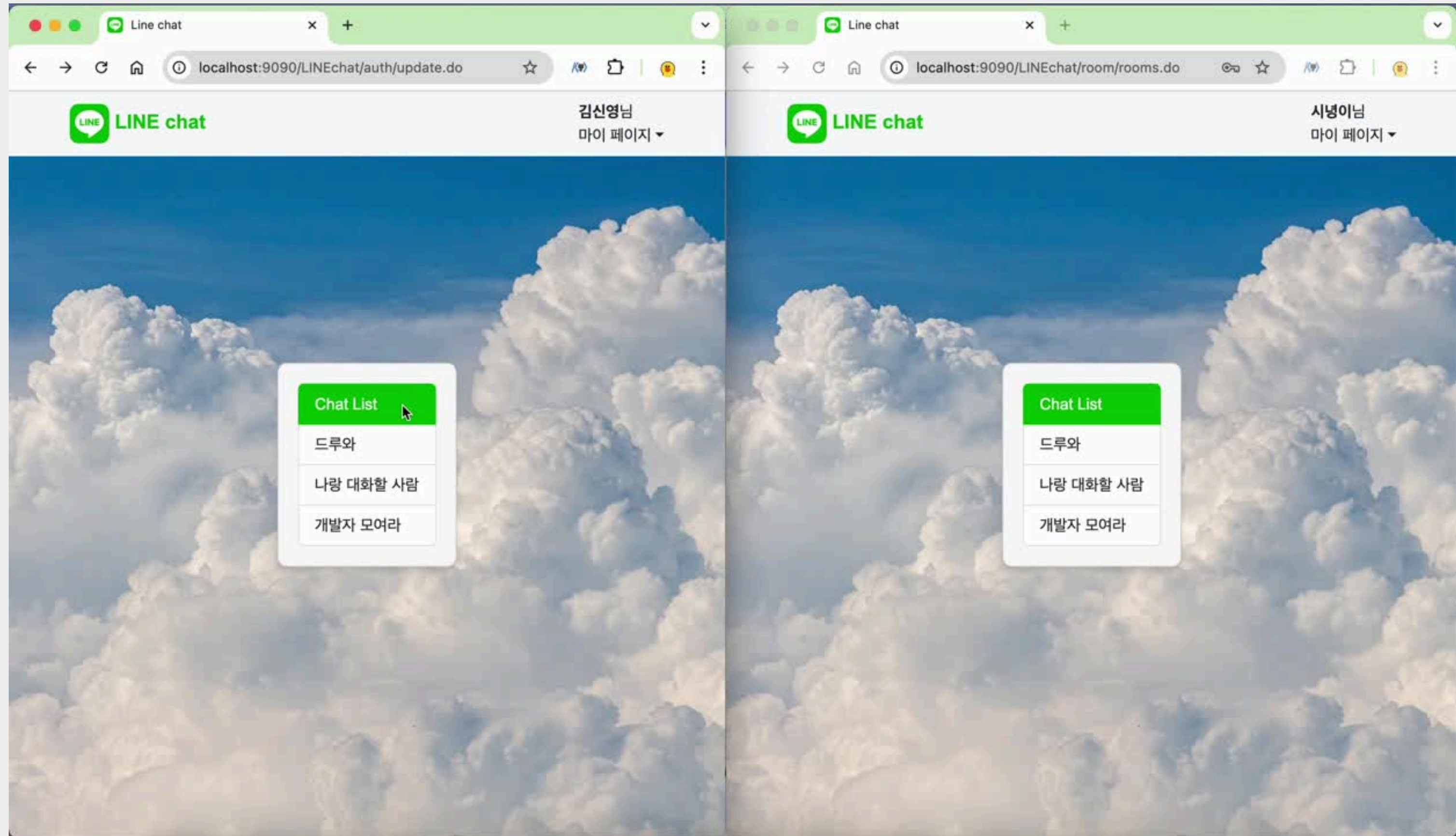
# \* 기능 소개

## 닉네임 수정



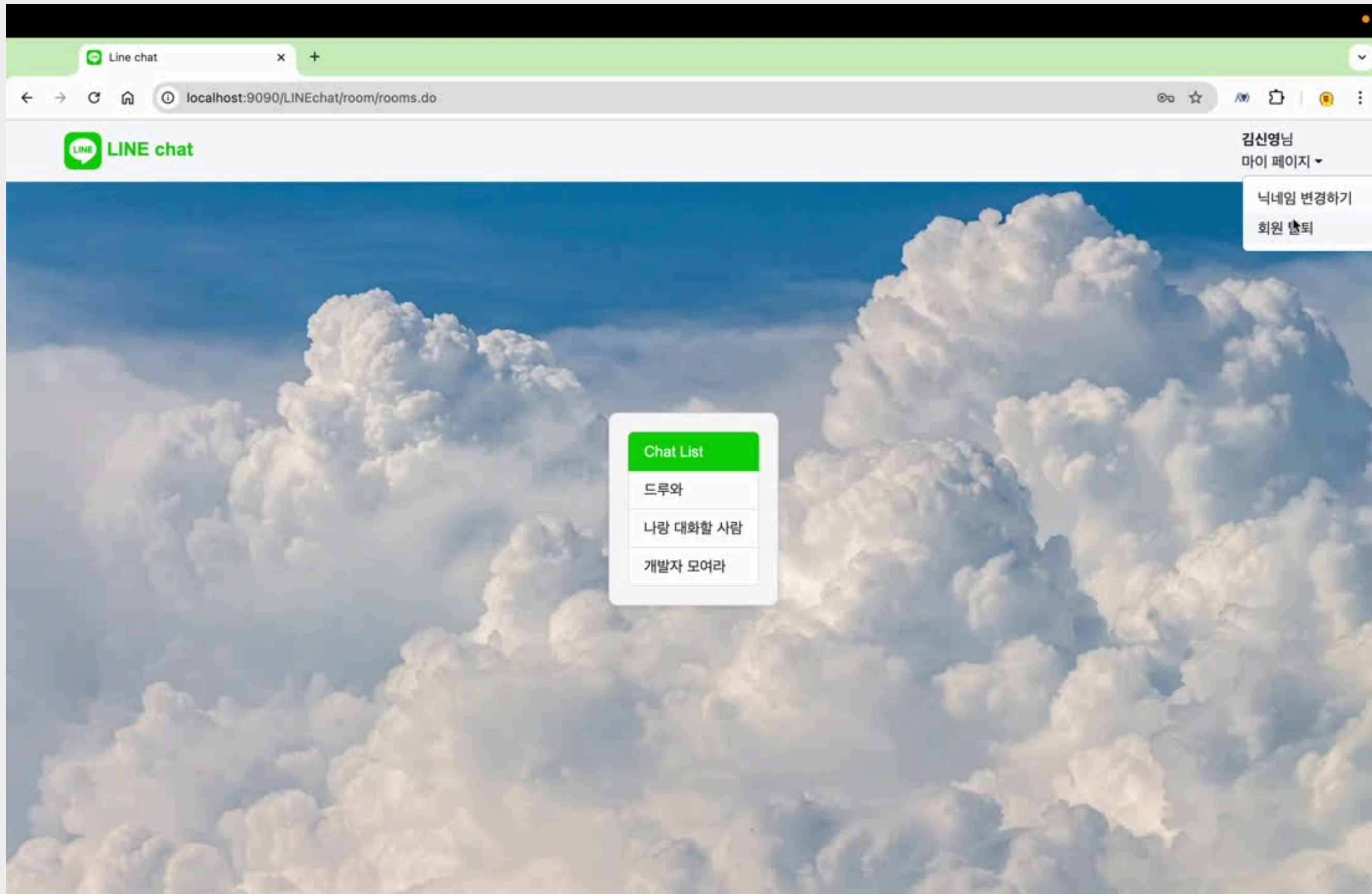


# \* 기능 소개 채팅 #입장알림 #퇴장알림 #인원수



# \* 기능 소개

회원 탈퇴





감사합니다.