can you do the real time updade code in java

**ChatGPT said:**

Yes! Here’s how you can implement **real-time online status updates** in Java using **WebSockets (Spring Boot)** and **MySQL**.

**1. Modify group\_members Table**

Since you've already provided your table, ensure you've added these columns:

sql

CopyEdit

ALTER TABLE group\_members

ADD COLUMN online\_status ENUM('online', 'offline') DEFAULT 'offline' AFTER `joined\_at`,

ADD COLUMN last\_active TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP AFTER `online\_status`;

**2. WebSocket Configuration in Spring Boot**

To enable real-time updates, we need **WebSockets**.

**Step 1: Add Dependencies (pom.xml)**

xml

CopyEdit

<dependency>

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

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

</dependency>

<dependency>

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

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

</dependency>

**3. Create WebSocket Configuration**

Create a **WebSocketConfig.java** file to enable WebSockets.

java

CopyEdit

import org.springframework.context.annotation.Configuration;

import org.springframework.web.socket.config.annotation.EnableWebSocket;

import org.springframework.web.socket.config.annotation.WebSocketConfigurer;

import org.springframework.web.socket.config.annotation.WebSocketHandlerRegistry;

@Configuration

@EnableWebSocket

public class WebSocketConfig implements WebSocketConfigurer {

@Override

public void registerWebSocketHandlers(WebSocketHandlerRegistry registry) {

registry.addHandler(new OnlineStatusHandler(), "/ws/online-status")

.setAllowedOrigins("\*");

}

}

**4. Create WebSocket Handler**

Create a **WebSocket Handler** that listens for online/offline status updates.

**OnlineStatusHandler.java**

java

CopyEdit

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

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

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.util.Map;

import java.util.concurrent.ConcurrentHashMap;

public class OnlineStatusHandler extends TextWebSocketHandler {

private static final String DB\_URL = "jdbc:mysql://localhost:3306/your\_database";

private static final String DB\_USER = "your\_user";

private static final String DB\_PASSWORD = "your\_password";

private static final Map<String, WebSocketSession> sessions = new ConcurrentHashMap<>();

@Override

public void afterConnectionEstablished(WebSocketSession session) throws Exception {

String userId = getUserIdFromSession(session);

updateOnlineStatus(userId, "online");

sessions.put(userId, session);

broadcastOnlineUsers();

}

@Override

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

String userId = getUserIdFromSession(session);

updateLastActive(userId);

}

@Override

public void afterConnectionClosed(WebSocketSession session, CloseStatus status) throws Exception {

String userId = getUserIdFromSession(session);

updateOnlineStatus(userId, "offline");

sessions.remove(userId);

broadcastOnlineUsers();

}

private String getUserIdFromSession(WebSocketSession session) {

return session.getUri().getQuery().split("=")[1]; // Example: ws://localhost:8080/ws/online-status?userId=5

}

private void updateOnlineStatus(String userId, String status) {

try (Connection conn = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD)) {

String sql = "UPDATE group\_members SET online\_status = ?, last\_active = NOW() WHERE user\_id = ?";

try (PreparedStatement stmt = conn.prepareStatement(sql)) {

stmt.setString(1, status);

stmt.setInt(2, Integer.parseInt(userId));

stmt.executeUpdate();

}

} catch (Exception e) {

e.printStackTrace();

}

}

private void updateLastActive(String userId) {

try (Connection conn = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD)) {

String sql = "UPDATE group\_members SET last\_active = NOW() WHERE user\_id = ?";

try (PreparedStatement stmt = conn.prepareStatement(sql)) {

stmt.setInt(1, Integer.parseInt(userId));

stmt.executeUpdate();

}

} catch (Exception e) {

e.printStackTrace();

}

}

private void broadcastOnlineUsers() {

try (Connection conn = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD)) {

String sql = "SELECT user\_id FROM group\_members WHERE online\_status = 'online'";

try (PreparedStatement stmt = conn.prepareStatement(sql);

var rs = stmt.executeQuery()) {

StringBuilder userList = new StringBuilder();

while (rs.next()) {

userList.append(rs.getInt("user\_id")).append(",");

}

TextMessage message = new TextMessage(userList.toString());

for (WebSocketSession session : sessions.values()) {

session.sendMessage(message);

}

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

**5. Frontend JavaScript to Connect to WebSocket**

javascript

CopyEdit

let userId = 5; // Change this dynamically

let socket = new WebSocket(`ws://localhost:8080/ws/online-status?userId=${userId}`);

socket.onopen = function () {

console.log("Connected to WebSocket");

};

socket.onmessage = function (event) {

let onlineUsers = event.data.split(",");

console.log("Online Users: ", onlineUsers);

};

socket.onclose = function () {

console.log("Disconnected from WebSocket");

};

// Send a ping every 30 seconds to update last\_active

setInterval(() => {

socket.send("ping");

}, 30000);

**6. Automatically Mark Inactive Users as Offline**

If a user is inactive for **5 minutes**, mark them as **offline**.

java

CopyEdit

import org.springframework.scheduling.annotation.Scheduled;

import org.springframework.stereotype.Component;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

@Component

public class InactiveUserScheduler {

private static final String DB\_URL = "jdbc:mysql://localhost:3306/your\_database";

private static final String DB\_USER = "your\_user";

private static final String DB\_PASSWORD = "your\_password";

@Scheduled(fixedRate = 300000) // Runs every 5 minutes

public void markInactiveUsersOffline() {

try (Connection conn = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD)) {

String sql = "UPDATE group\_members SET online\_status = 'offline' WHERE last\_active < NOW() - INTERVAL 5 MINUTE";

try (PreparedStatement stmt = conn.prepareStatement(sql)) {

stmt.executeUpdate();

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

**7. Run and Test**

1. **Start Spring Boot** (mvn spring-boot:run).
2. **Open WebSockets in the browser console**:

javascript

CopyEdit

let socket = new WebSocket("ws://localhost:8080/ws/online-status?userId=5");

1. **Check database** to see if online\_status updates when users connect and disconnect.

**Final Steps**

✅ **Enable WebSockets** with WebSocketConfig.java  
✅ **Track user online status in group\_members table**  
✅ **Use OnlineStatusHandler.java for real-time updates**  
✅ **Mark inactive users as offline using a scheduler**  
✅ **Frontend WebSocket connection for real-time updates**

**Next Steps**

If you want me to tweak the code for your exact framework (Spring Boot, JavaFX, or something else), let me know! 🚀