

# 고급소프트웨어실습1 9주차 과제

컴퓨터공학 20172141 김미소

## 1. Websocket & socket.io 통신 방식

WebSocket은 기존의 요청-응답 관계 방식에서 벗어나 소켓을 이용하여 자유롭게 데이터를 주고 받을 수 있다. 먼저 클라이언트에 의해 요청을 받는 방식이 아니라 서버가 내용을 클라이언트에 보내는 표준화된 방식을 사용한다. 또 연결이 유지된 상태에서 메시지들이 오갈 수 있다. 이 프로토콜을 사용하기 위해서는 브라우저와 웹 서버가 WebSocket 기능을 지원해야한다. 브라우저가 "Upgrade: WebSocket" 헤더와 함께 랜덤하게 생성한 키를 서버에 보내면 웹 서버는 이 키를 바탕으로 토큰을 생성한 후 브라우저에 돌려준다. WebSocket 프로토콜은 하나의 80번 포트 TCP 커넥션을 이용하고, 별도의 헤더로 논리적인 데이터 흐름 단위를 이용하여 여러 개의 커넥션을 맺는 효과를 내는 방식을 사용한다. WebSocket 프로토콜은 HTTP에 비해 적은 오버헤드를 가진다.

Socket.io는 자바스크립트를 이용하여 브라우저 종류에 상관없이 실시간 웹을 구현할 수 있도록 한 기술이다. Socket.io는 WebSocket, FlashSocket, AJAX Long Polling, AJAX Multi part Streaming, IFrame, JSONP Polling을 하나의 API로 추상화한 것으로, 브라우저와 웹 서버의 종류와 버전을 파악하여 가장 적합한 기술을 선택하여 사용하는 방식이다. Socket.io는 Node.js 모듈로서 개발자가 구현 방법을 잘 알지 못해도 사용할 수 있다. io.Socket()을호출하면 웹 서버와 연결하고 서버로부터 sessionid 정보와 timeout 정보를 받고, 브라우저의 WebSocket 지원 여부, FlashSocket 지원 여부를 보내고 크로스 도메인 설정 정보 등을 주고 받은 후 적합한 실시간 웹 방식을 선택한다.

## 2. Websocket/socket.io를 통한 채팅 프로그램 구현 방식

간단하게 말하자면, 프론트엔드 혹은 백엔드에서 emit으로 데이터를 보내면 on으로 받으면 된다. Server.js에 socket 라이브러리 입력을 하고 server와 socket을 연결한다. 서버에서 username이나 message와 같은 데이터를 받았을 때 어떤 동작을 할지 코드를 작성하여 선언할 수 있다. 클라이언트부분에서 소켓 연결 선언은 컴포넌트 함수 안에 하는 것이 아니라 컴포넌트 함수 밖에 선언하여 항상 연결되어 있을 수 있도록 해야 한다. 사용자가 username이나 message를 입력했을 때 데이터를 state에 저장하는 것뿐만 아니라 소켓을 통하여 서버에 전달될 수 있도록 emit 해주어야 한다. 그리고 서버에 전송된 메시지를 채팅창에 출력하기 위하여 useEffect 함수 안에서 on을 통해 서버에서 데이터를 받아오고 로컬 state에 갱신하여야 한다. 이러한 방법으로 소켓을 통해 채팅 프로그램을 구현할 수 있다.