**고급소프트웨어실습1: 9주차 과제**

**20182204 한찬희**

**프로그램 소개**

React를 활용하여 Chat App을 구현한다.

**프로그램 구동 방법**

1. 처음 받았을 때 npm install -> npm ci 순으로 패키지들을 설치해준다.
2. npx yarn start로 안전하게 프로그램을 실행시키어준다.
3. Sending Message에 내용을 입력하여 새로운 메세지를 Log에 남긴다.
4. Chat Log에서는 지금까지 남겨진 Log들을 출력해준다.
5. 검색창에 찾고자 하는 키워드를 입력하고 Enter을 입력하면 해당 채팅 내용이 빨간색으로 표시된다.

**websocket과 socket.io 통신 방식에 대해 자세히 조사한 후 이에 대해 기술**

* **websocket :** 웹 소켓이란 두 프로그램 간의 메시지를 교환하기 위한 통신 방법 중 하나

특징

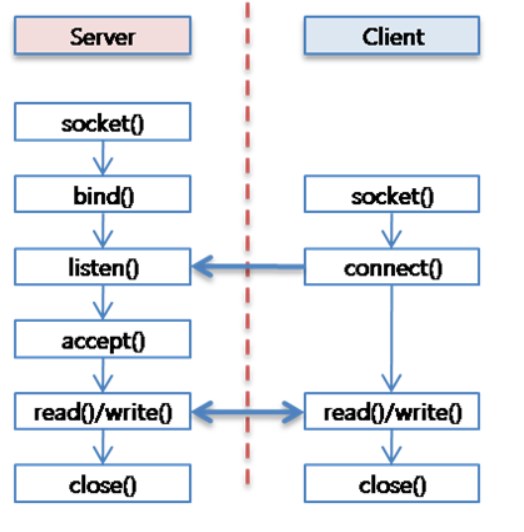
1. 양방향 통신 : 데이터 송수신을 동시에 처리할 수 있는 통신 방법, 클라이언트와 서버가 서로에게 원할 때 데이터를 주고 받을 수 있다.
2. 실시간 네트워킹 : 웹 환경에서 연속된 데이터를 빠르게 노출하는 서비스에서 여러 단말기에 빠르게 데이터를 교환시 사용.

동작 방법

1. http, https를 이용해 핸드 쉐이킹
2. ws, wss 프로토콜을 이용해 데이터 전송.

* socket.io : HTML5 이전의 기술로 구현된 서비스에서 웹 소켓처럼 사용할 수 있도록 도와주는 기술. Javascript를 이용하여 브라우저 종류에 상관없이 실시간 웹을 구현한다. 즉, 브라우저와 웹 서버의 종류와 버전을 파악하여 가장 적합한 기술을 선택하여 사용하는 방식

**실제로 websocket/socket.io 통신 방식을 사용해서 실제 채팅 프로  
그램을 구현하게 됐을 때의 전체적인 구현 방식 기술**



1. Server쪽에서는 app, server, cors, io 등을 미리 bind 해 놓고 Client의 요청을 기다린다.
2. Client 측에서 socket을 서버쪽으로 하여 연결한다. (socket = io.connect(서버주소))
3. Client 측에서 소켓을 가지고 메시지를 서버로 전송한다. (socket.emit(데이터))
4. Server 측에서 열어놓은 소켓으로 온 메시지를 받는다.(io.on(데이터))
5. Server 측에서 읽은 데이터를 처리하고 이후 지속적으로 socket을 이용해 실시간 통신을 하게된다.

그렇기에 서버와 클라이언트 별로 소켓을 만든다. 서버는 listen을 상시 가동하여 client의socket으으부터 socket.emit된 데이터를 기다린다. 서버에서는 socket.on으로 데이터를 받아 해당 데이터를 처리한다. 첫 핸드 쉐이킹 과정이 끝나면 이후 연결된 소켓을 사용해 상호 실시간 통신을 하게 된다.