



# Project6 (Realtime Drawing)

## 실시간 그림판 만들기(230522)

NodeJS 수강 완료 후 앞서 배운 것들(HTML, CSS, JS, React, NodeJS, express, mongoDB) 활용

### 개발일지

CURL로 제작

canvas, 마우스 위치 이벤트, express라이브러리, socket.io, mongoDB 기술을 익히기 위해 시작

클라이언트(React)의 그림판에서 마우스 위치 이벤트(그림)를 실시간으로 공유할 수 있게 만든다.

⇒ React Router사용, canvas사용

서버(Nodejs)에서 해당 url로 입장한 모든 유저에게 그림을 공유해준다.

DB에 유저정보(이름), 방 정보(방id, **해당 방에 그렸던 그림**, 방 접속 유저) 저장

방에 있던 유저가 모두 나가면 방은 사라짐

⇒ socket.io, express, mongoDB사용

참고 : <https://collabio-kriziu.herokuapp.com/>

- 시작화면
- 방 입장 후

# Collabio

Real-time whiteboard

Enter your name

Username...

Enter room id

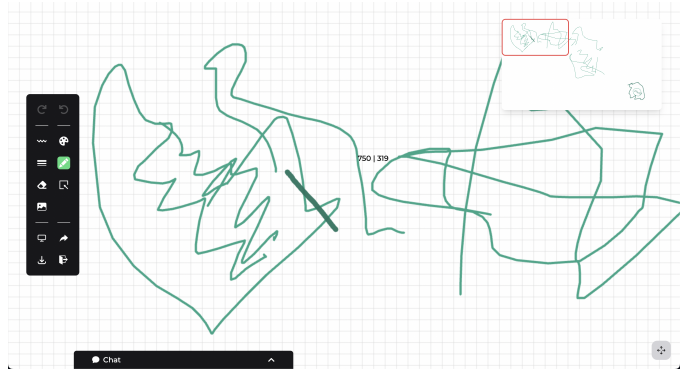
Room id...

Join

or

Create new room

Create



## 컴포넌트 설계

### App컴포넌트(다크모드 지원)

- **방입장 컴포넌트(React, NodeJS, Express, MongoDB)**

- 유저이름
- 방id
- 방입장, 방 만들기(react router)

- **그림판 컴포넌트(격자 표시)**

- 그림판(socket.io)
- **그림도구 컴포넌트**
  - 펜, 지우개, 펜 두께, 색 고르기
  - 사진 업로드(사진 위에 그림 가능)
  - 그림 저장, 방 나가기, 초대하기

- **채팅 컴포넌트**

- 유저 채팅(socket.io)
- 접속중인 유저, 유저 수 표시



## 기능 설계

첫 시작화면은 항상 방 입장(start) 컴포넌트여야한다.

유저가 새로운 방을 생성할 수 있기 때문이다.



### 방 생성하기 버튼으로 form 제출 후(fe)

- 제출된 id를 가진 방이 존재하는지 확인
  - 방 id가 중복되는 경우 아이디 다시 입력
  - 해당 id가 존재하지 않으면(server,db) ⇒ 해당 id로 새로운 방 생성, 닉네임 저장(db, fe).

### 방 참가하기 버튼으로 form 제출 후(fe)

- 입력한 방 id가 존재할 경우(server,db) ⇒ 해당 방으로 입장(fe).
- 해당 id가 존재하지 않으면(server,db) ⇒ 아이디 다시 입력

- 방에 있던 유저가 모두 나가면 방은 사라짐
- 서버, DB에서 확인 후 리액트에서 라우팅
- DB에 유저정보(이름), 방 정보(방id, 해당 방에 그렸던 그림, 방 접속 유저) 저장
- 초대 링크를 받은 사람은 닉네임만 생성해도 초대받은 방으로 입장할 수 있다(fe, server, db).
- 초대 링크는 복사 형식(모달)

### • 그림도구 컴포넌트

- 펜, 지우개, 펜 두께, 색 고르기
- 사진 업로드(사진 위에 그림 가능)
- 그림 저장, 방 나가기, 초대하기

### • 채팅 컴포넌트

- 유저 채팅(socket.io)

◦ 접속중인 유저, 유저 수 표시

▼ !!!!!ERRORS!!!!!!



## 시작부터 라이브러리 설치 오류에 부딪힘(2시간 삽질)

styled-components를 사용하려고 명령어를 사용중에

```
npm install styled-components 가 아닌  
npm install styled-component 로 작성해서
```

잘못된 줄 알고 설치 도중에 중단 후 다시 설치하려 했는데  
이런 에러가 생겼다.

```
shinyouchang@sin-yuchang-ui-MacBookAir pj6 % npm install styled-compone  
nts  
npm ERR! Cannot read properties of null (reading 'edgesOut')  
  
npm ERR! A complete log of this run can be found in:  
npm ERR!     /Users/shinyouchang/.npm/_logs/2023-05-22T05_46_16_380Z-de  
bug-0.log
```

Nodejs, npm을 전체 삭제하고, 시스템 리부팅도 해보고 할 수 있  
는것 모두 시도해봄

⇒ 오류 수정 실패

구글링으로 해결 했다.

<https://luminous24.tistory.com/269>

### 문제 해결

```
//Use V5,  
npm install styled-components@5.3.10  
//Use yarn  
yarn install styled-components  
//To use the beta version  
npm install styled-components@latest
```

! **Error: [Immer] An immer producer returned a new value *and* modified its draft. Either return a new value *or* modify the draft.**

검색 결과.

<https://velog.io/@ohy9205/redux-toolkit-Immer-An-immer-producer-returned-a-new-value-and-modified-its-draft.-Either-return-a-new-value-or-modify-the-draft>

redux-toolkit createSlice로 만든 action을 컴포넌트에서 호출할 때 발생

state 변경 시 새 값을 반환하면서 원본도 수정했기 때문에 발생하는 에러

redux-toolkit의 immer사용 규칙을 보면 원본을 변경하던가 새 값을 리턴하던가 둘 중 하나만 해야함

chatgpt 결과.

오류의 원인은 다음 코드 블록에서 발생할 수 있습니다:

```
dispatch(joinRoom(result));
```

해당 코드는 `dispatch` 함수를 호출하여 `joinRoom` 액션을 디스패치합니다.

그러나 `joinRoom` 액션 생성자에서 값을 수정하고 새로운 값을 반환하지 않은 것으로 보입니다.

Immer는 기본적으로 불변성을 유지하도록 설계되었기 때문에, 값을 수정하는 동시에 새로운 값을 반환하는 것은 규칙을 위반하는 것입니다.

해당 부분을 수정하여 값을 반환하도록 변경할 수 있습니다:

```
dispatch(joinRoom(result.data));
```

위 코드에서 `result.data`를 반환하도록 변경하면 오류가 해결됩니다.

이렇게 하면 `joinRoom` 액션의 페이로드로 `result.data` 값을 전달하게 됩니다.

---



## button type="submit" 문제 ⇒ 2시간 반 삽질

axios로 데이터를 주고받는데는 전혀 문제가 없었고,

페이지 라우팅도 문제가 없었지만

이상하게 방 참가하기와 방 생성하기 버튼을 눌렀을 때 동작해야하는 이벤트 핸들러가 전혀 먹히지 않았다.

chatgpt에 코드를 모두 보여줘도 이상한 말만 늘어났다.

문제는 방 참가하기와 방 생성하기 버튼에 있었다.

**type이 submit으로 되어있어서 그랬다.**

**type="submit"으로 설정된 버튼은 클릭하면 기본 동작인 폼 제출 (submit)을 수행해서**

**페이지를 다시 로드하거나 새로고침하는 동작을 수행하는데,**

**이 때 이벤트 핸들러가 실행되지 않을 수 있다는 사실을 모르고 있었다.**

브라우저 콘솔에 아무런 오류도 뜨지 않고, 서버 콘솔에는 받아온 데이터가 잘 뜨길래 더더욱 문제를 잡을 수 없었다.

방 생성하기 이벤트 핸들러에서

```
const createRoomHandler = async () => {
  axios
    .post("http://localhost:8080/create", {
      body: {
        drawid: enteredDrawid,
        nickname: enteredNickname,
      },
    }) => {
      // 여기서 이미 해당 id로 존재하는지 구별 해야함.
      // 없는 방 만들때 == response.data.insertedId
      // 이미 존재하는 방일때 == response.data.roomid
      console.log("create", response.data);
      if (response.data === "") {
        alert("해당 ID로 된 방이 이미 존재합니다. 다른 ID로 생성해 주세요.");
      } else {
        console.log(response.data);
        dispatch(joinRoomInfo(response.data));
        navigate("/draw");
      }
    }
}
```



```
.catch((error) => {  
  console.log("create 요청실패");  
  console.log(error);  
});  
};
```

앞으로 버튼의 타입을 처음부터 제대로 설정해야겠다.

페이지가 새로고침 되는 것 같다면 form이 제출되는 것은 아닌지 체크할 것.

form 태그에 대해 좀 더 공부해야겠다.

이벤트 핸들러를 실행하려면 `type="button"` 으로 설정,

폼 제출을 위해 버튼을 사용하려면 `type="submit"` 으로 설정할 것.

---



#### 4시간 삽질

**place-content: flex-end;**

draw 페이지의 레이아웃을 거의 다 완성하고,

채팅 컴포넌트(.bubbles) 안에 채팅이 설정한 height을 넘어가면 스크롤이 생기게 하려고 했지만, 어떤 이유에선지 생기질 않았다.

구글링도 해보고 gpt에도 물어 봤지만 명확한 답이 없었다.

혹시나 body에 넣은 overflow:hidden 속성이 문제 일까봐 썼다 지웠다만 500번은 한 것 같다.

부모 태그들의 높이도 모두 다 조절 해보고, html태그에 overflow:scroll도 넣어보고 다 봤다.

어쩔 수 없이 있는 속성들을 모두 하나씩 검사 하다가 **place-content: flex-end;**를 주석처리하자 scroll이 되기 시작했다.

(index.css)

```
body {
  margin: 0;
  overflow: hidden;
}
```

(Chat.module.css)

```
.bubbles {
  height: 451px;
  overflow: scroll;
  display: flex;
  flex-direction: column;
  border: 1px solid rgba(166, 166, 166, 0.345);
  border-radius: 15px 15px 0px 0px;
  /* place-content: flex-end; */
  background-color: white;
  box-shadow: 5px 5px 15px -10px inset;
}
```

```
import React from "react";

import classes from "./Chat.module.css";

const Chat = () => {
  return (
    <div className={classes.chatbox}>
      <div className={classes.chatting}>
        <div className={classes.bubbles}>
          <div className={classes.bubble}>상대방 채팅</div>
          <div className={` ${classes.bubble} ${classes.bubbleMe}`}>내 채팅</div>
          <div className={classes.bubble}>상대방 채팅</div>
        </div>
        <div className={classes.send}>
```

```

        <input type="text" />
        <button>></button>
      </div>
    </div>
  </div>
);
};

export default Chat;

```



이제 잘됨

## cors error

```

✖ Access to XMLHttpRequest at 'http://localhost:8080/socket.io/?EIO=4&transport=polling&t=0Xccobg' from origin 'http://localhost:3000' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.
✖ ▶ GET http://localhost:8080/socket.io/?EIO=4&transport=polling&t=0Xccobg polling.js:307 ⚡
net::ERR_FAILED 200 (OK)

```



## cors에러

```
const io = new Server(http, {  
  cors: {  
    // 허용할 도메인 주소  
    origin: "http://localhost:3000",  
    // 허용할 HTTP 메서드  
    methods: ["GET", "POST"],  
  },  
});
```