Smooth팀 frontend 김민지 코드리뷰 가이드

## [함께 보시면 좋은 문서]

https://github.com/stove-smooth/sgs-smooth/blob/develop/src/frontend/web/README.md

## [프로젝트 소개]

저희 Smooth 팀은 실시간 채팅, 음성, 화상 통화를 지원하는 서비스인 discord를 클론 코딩하였습니다. 그리고 저희 팀의 목표와 취향을 고려하여 원하는 서비스 기능을 조합하여 Smooth 팀만의 실시간 채팅, 음성, 화상 통화 서비스를 설계하였습니다.

저희 Smooth 팀이 클론한 디스코드의 기능은 크게 사용자가 목적이 맞는 다른 사람들과 채팅/통화를 할 수 있는 커뮤니티와 커뮤니티를 벗어나서 원하는 사람들과 1:1 혹은 그룹으로 채팅/통화를 할 수 있는 Direct Message(DM)으로 나뉩니다.

커뮤니티에서는 소속된 멤버들과 함께 카테고리와 채널을 만들어 커뮤니티를 취향대로 커스텀할 수 있습니다.

## [프로젝트 용어 소개]

디스코드의 커뮤니티는 원래 명칭이 서버인데 개발을 진행하는 중, 명칭이 이해가 힘들어 중간에 커뮤니티로 바꾸게 되었습니다. DM과 커뮤니티를 여러 이름으로 쓴 것을 계속해서 통일시키고 있습니다.

| | |
|---|---|
| 커뮤니티, 서버 => | 커뮤니티 |
| DM, room, directMessage => | DM |
| Voice,participant (store 폴더 내부 파일) => | webRTC 관리를 위한 파일 |

## [화면 구성]

Routes(src/routes/index.js) 파일과 함께 저희 서비스의 화면 구성을 말씀드리겠습니다.

라우트는 크게 로그인 전/후로 나뉩니다.

**<로그인 전>**

- 로그인/회원가입 기능 화면 (LoginPage, RegisterPage)

- 초대 링크로 들어올 경우, 처리를 하기 위한 화면 (InvitePage)

**<로그인 후>**

- 로그인시 진입하는 화면 (MainPage)
- 나의 Direct Message List, 친구의 활동을 볼 수 있는 화면(MyPage)
- Direct Message 내역을 보고, 채팅/통화를 할 수 있는 화면(PrivateDMPage)
- 커뮤니티 사람들과 채팅/통화를 하며, 커뮤니티를 커스텀할 수 있는 커뮤니티 화면 (CommunityWelcomePage, CommunityPage)
- 개인 정보 설정 화면(UserSettingPage)

## [화면의 구성요소 및 기능 소개]

라우트 파일을 기반으로 화면들의 구성 요소 및 기능에 대해 소개하겠습니다.

### [LoginPage, RegisterPage]

로그인/가입을 담당하는 화면입니다. 이 프로젝트에서는 localStorage를 이용하여 accesstoken과 refreshtoken을 관리합니다.

### [InvitePage]

커뮤니티에서는 다른 사람을 초대하기 위해 '초대장'이라는 기능을 사용합니다. 초대장 링크를 주소창에 입력하면 이 화면을 통해 로그인이 되어있을 경우엔 서버로, 로그아웃 시에는 로그인을 해서 입장할 수 있도록 합니다.

### [MainPage]

로그인 시 제일 먼저 입장하는 화면입니다. 프로젝트 초반, socket 연결 시점 문제로 구독한 메시지를 받을 수 없는 문제가 있었는데 이 문제를 해결하기 위해 트리 구조를 사용하여 로그인 후의 화면들은 모두 MainPage의 자식으로 두었습니다.

MainPage는 navigationbar와 자식 page들이 보이는 router-view, 모달창으로 구성됩니다. 그리고 socket 연결과 알림 토큰을 받아오는 기능을 담당합니다.

### [MyPage]

- Direct Message 목록(friends-side-bar)
- 마이크, 헤드셋, 개인 설정 조절 가능한 사용자 영역(user-section)

- 친구 요청을 할 수 있는 영역(friends-new-add)
- 온라인 상태인 친구, 모든 친구, 친구 요청 대기/친구 응답 대기인 사람, 차단된 친구 등 조건에 따른 친구들의 목록을 볼 수 있는 영역(friends-state-list)
- 보이는 친구들의 조건을 조절할 수 있는 영역(friends-state-menu-bar)

로 화면 및 기능이 구성됩니다.

## [PrivateDMPage]

Direct message 소속 멤버들과 채팅 및 통화를 할 수 있습니다.

**[voice-sharing-area]**

- ◆ 통화 시작 시 이 컴포넌트가 보입니다. vuex에서 관리하는 음성 참가자를 감지하여 변화가 있을 때마다 참가자 비디오를 화면에 새로 그려줍니다. Community에서 통화 시에도 이 컴포넌트를 함께 사용합니다.
- ◆ User-section과 voice-sharing-area에 있는 장치 제어를 통해 비디오, 마이크, 헤드셋을 on/off 할 수 있습니다.

**[dm-activity-area]**

- ◆ Direct message 입장 시 채팅 기록을 볼 수 있고 실시간 채팅을 할 수 있는 컴포넌트입니다. 실시간 채팅/수정/답장이 이루어집니다. 채팅 입장 시 상태를 서버에 알렸고, 후에 알림을 적용할 계획입니다.
- ◆ 이미지 채팅을 과거에 보냈을 경우 처음 채팅 내용을 불러올 때 스크롤이 맨 아래로 가지 않는 문제가 있었습니다. 프론트엔드 코드리뷰 요청을 통해 이미지가 로딩되는 시점이 다르다는 사실을 알았고, 피드백을 적용하여 그 문제를 해결하였습니다.
- ◆ 타이핑 상태를 서버에 전송하고 받아 실시간으로 채팅을 작성 중인 사용자가 화면에 뜨게끔 하였습니다.

## [CommunityPage & CommunityWelcomePage]

커뮤니티 입장 시 CommunityWelcomePage로 먼저 이동합니다. 커뮤니티 정보(카테고리, 채널)을 보고 이동할 채팅 채널이 있다면 CommunityPage로 이동합니다. 만약, 이동할 채팅 채널이 없다면(음성 채널만 존재 혹은 채널이 존재하지 않음) 그대로 있습니다.

**[**community-side-bar**]**

- ◆ 채널 이동, 채널 커스터마이징, 음성 참여자 목록 확인이 가능합니다.

[community-activity-area][voice-sharing-area]

- DM 채팅/통화와 동일합니다.

**[UserSettingPage]**

간단한 프로필을 수정할 수 있습니다.

## [참고 사항]

Kurento group call javascript 데모를 vue에 적용하는 부분은

https://github.com/ritajeong/focus를 참고하였습니다.

Vue에 처음 적용할 때 this 문제와 음성 통화 참가자 관리 문제 등 여러 문제가 발생하여 작동이 되지 않았습니다.

그때 이 코드를 보고 vuex에 예제를 javascript 그대로 적용하여 this 문제를 해결할 수 있었고 참가자 관리 및 발생한 여러 문제를 해결할 수 있었습니다.