

## Side Panel Direct Message 기본 UI

기본 UI

```
renderDirectMessages = (users) => users.length > 0 &&
  users.map(user => (
    <li
      key={user.uid}
    >
      # {user.name}
    </li>
  ));

render() {
  const { users } = this.state;

  return (
    <>
      <span>
        DIRECT MESSAGES ({users.length})
      </span>
      <ul style={{ listStyleType: 'none', padding: '0' }}>
        {this.renderDirectMessages(users)}
      </ul>
    </>
  )
}
```

Direct Message를 위한 User 목록 가져오기

user 목록 가져오기  
addUsersListener

```
componentDidMount() {  
  if (this.state.user) {  
    this.addUsersListeners(this.state.user.uid);  
  }  
}  
  
addUsersListeners = currentUserUid => {  
  let usersArray = [];  
  const { usersRef } = this.state;  
  usersRef.on("child_added", DataSnapshot => {  
    if (currentUserUid !== DataSnapshot.key) {  
      let user = DataSnapshot.val();  
      user["uid"] = DataSnapshot.key;  
      user["status"] = "offline";  
      usersArray.push(user);  
      this.setState({ users: usersArray });  
    }  
  })  
}
```

John Ahn, 16 days ago • direct message

Direct Message 목록 클릭

유저 이름 클릭 시  
해당 유저와 대화 할  
대화방 아이디 생성

```
changeChatRoom = user => {  
  const chatRoomId = this.getChatRoomId(user.uid);  
  const chatRoomData = {  
    id: chatRoomId,  
    name: user.name  
  };  
};  
  
getChatRoomId = userId => {  
  const currentUserId = this.props.user.uid;  
  //나와 다른 사람의 채팅방 ID가 유니크하면서 내가 대화를 할 때 그  
  //그러기 위해서 나의 아이디와 상대방의 아이디를 이용해서 방 아이디  
  //userId < currentUserId 와 같은 로직을 이용해서 방 아이디  
  return userId < currentUserId  
    ? `${userId}/${currentUserId}`  
    : `${currentUserId}/${userId}`;  
};
```

A:1234      B:5678

1234/5678

currentChatRoom을 클  
릭한 방으로 변경

```
changeChatRoom = user => {  
  const chatRoomId = this.getChatRoomId(user.uid);  
  const chatRoomData = {  
    id: chatRoomId,  
    name: user.name  
  };  
  this.props.dispatch(setCurrentChatRoom(chatRoomData));  
};
```

클릭한 방이 private 방임을 알려주기

```
this.props.dispatch(setPrivateChatRoom(true));
```

```
case SET_PRIVATE_CHAT_ROOM:  
  return {  
    ...state,  
    isPrivateChatRoom: action.payload  
  }  
}
```

클릭한 방을 Active 방으로 표시해주기

```
this.setActiveChatRoom(user.uid);
```

```
state = {  
  usersRef: firebase.database().ref("users"),  
  users: [],  
  activeChatRoom: "",  
}
```

```
setActiveChatRoom = userId => {  
  this.setState({ activeChatRoom: userId });  
};
```

## Private, Public 채팅방 구분해주기

private과 public 아이콘  
변화시켜주기

```
{
  isPrivateChatRoom ?
    <FaLock style={{ marginBottom: '10px' }} />
    :
    <FaLockOpen style={{ marginBottom: '10px' }} />
}
{" "}
{chatRoom && chatRoom.name}
```

public 채팅 방으로 갈 때  
setPrivateChatRoom  
false로 바꿔주기

```
const changeChatRoom = (chatRoom) => {
  dispatch(setCurrentChatRoom(chatRoom));
  setActiveChatRoomId(chatRoom.id);
  dispatch(setPrivateChatRoom(false));
}
```

private에서 올리는 파일  
들은 public 파일과 다른  
경로로 올리기

```
const filePath = `message/public/${file.name}`;
```

```
const filePath = `${getPath()}/${file.name}`;
```

```
const getPath = () => {
  if (isPrivateChatRoom) {
    return `message/private/${chatRoom.id}`;
  } else {
    return `message/public/`;
  }
}
```

```
        return message/public ;  
    }  
};
```