

## 20170515 Week 3 Progress Report

### 1. 저번 주 구현 수정 사항 및 이번 주에 구현한 behaviors 정리

#### (1) 사가 함수 대격변 - 저번 주 구현에 있었던 문제점 수정 (배재복)

1. 기존에 사용하던 redux-saga-router 패키지 대신 window.localStorage를 이용한 구현
2. URL은 바뀌지만 reload되지 않던 문제 해결 & Generator is already running 문제 해결
3. 수정 사항 및 localStorage를 쉽게 사용하기 위한 comment 작업

#### (2) Chatting 백엔드 및 프론트엔드 구현 - 오픈 채팅방 형식

##### 1. Chatting 백엔드

- \* Chatting 기능을 구현하기 위한 백엔드 모델 설계 (김다윤)
- \* Chatting 백엔드 구현 (김다윤)
  - ChatRoom, ChatUser, Text (models.py)
  - ChatRoomSerializer, ChatUserSerializer, TextSerializer (serializers.py)
  - chatroom\_list, chatroom\_detail, chatuser\_list, chatuser, text\_list, text (views.py)
- \* Chatting 백엔드 관련 추가 사항 구현 (유가온)
  - 사용자가 참여한 채팅방과 참여하지 않는 채팅방을 따로 보여주는 view function 구현 및 URL 추가
  - NowChatSerializer, NonChatSerializer (serializers.py)
  - user\_nowchat, user\_nonchat, user\_detail (views.py)

##### 2. Chatting 프론트엔드 (안단태)

- \* Chatting에 관련된 action, reducer의 state 새로 정의
- \* /room/, /chatting/{id}, /create\_room/ 에서 작동하는 saga 함수 구현
- \* Chatting 관련 component 렌더링
  - 본인이 참여한 방과 참여하지 않은 방을 분리해서 보여주지 않음
- \* 깔끔하지 않았던 프론트엔드 코드의 comment 정리

#### (3) 2주차에 끝내지 못했던 Edit 및 Delete 구현 (김다윤)

1. Edit 및 Delete 관련 action 및 saga 함수 정의
2. Edit 페이지 component 렌더링

- \* EditArticle, EditArticlePage

#### (4) 프론트엔드 / 백엔드 테스트코드

##### 1. 2주차 프론트엔드 테스트코드 (배재복)

- \* 로그인 / 로그아웃 / 회원가입 테스트
- \* 메인 페이지 렌더링 / 글 작성 / 좋아요 / 수정 / 답글 / 삭제
- \* 디테일 페이지 렌더링 / 좋아요 / 답글 / 삭제
- \* 다른 사용자가 글 수정 / 삭제 시도 시 경고 메시지 확인

##### 2. 3주차 백엔드 테스트코드 (유가온)

- \* 3주차 프론트엔드 테스트코드 미구현
- \* chatroom\_list GET & POST / chatroom\_detail GET & DELETE
- \* text GET & POST / chatuser GET & POST
- \* chat의 user\_num 확인 / user\_list GET
- \* 테스트로 만들어진 모든 데이터 삭제

#### (5) 그 외의 사항

1. 백엔드 감수 (유가온)
  - \* 백엔드 구현 사항 확인 및 피드백

### 2. 새로 구현 및 수정한 API 및 테스트코드

#### (1) 백엔드 코드

- \* models.py: Chat, ChatUser, Text 모델 추가
- \* serializers.py: ChatRoomSerializer, ChatUserSerializer, TextSerializer, NowChatSerializer, NonChatSerializer 추가
- \* urls.py: chatroom/, chatroom/{id}/, chatuser/, chatroom/{id}/user/, text/, chatroom/{id}/text/, users/{id}/nowchat/, users/{id}/nonchat/ 추가
- \* views.py: chatroom\_list, chatroom\_detail, chatuser\_list, chatuser, text\_list, text, user\_nowchat, user\_nonchat, user\_detail 추가

(2) 프론트엔드 코드

- \* ChattingRoomPage: 채팅 목록 페이지. 각 채팅방에 Join / Chat, 메인으로 돌아가기, 새 채팅방 만들기.
- \* CreateRoomPage: 채팅방 생성 페이지. 채팅방 명 입력 및 채팅방 만들기 버튼 누르기
- \* ChattingPage: 채팅방 페이지. 채팅내용 작성하기. 채팅 내용 보내기.
- \* EditArticlePage: 글 편집페이지. 작성했던 글 편집하기. 글 수정하기 버튼 누르기.

(3) 테스트코드

- \* backend\_test.py => backend\_test1.py
- \* backend\_test2.py: behaviors정리의 4-2 확인
- \* backend.py & frontend.py: myTest.py에서 사용할 백엔드에 request를 보내는 함수 / 프론트엔드 테스트 함수가 정의된 파일
- \* myTest.py: behaviors정리의 4-1 확인

### 3. 제출 서버 확인 및 테스트 스크립트 실행 방법

#### (1) 제출 서버

- \* 프론트엔드: <http://wlxyzlw.iptime.org:3000/>
- \* 백엔드: <http://wlxyzlw.iptime.org:8000/>

#### (2) 테스트코드 실행 방법

- \* 백엔드 테스트코드 (원격 서버에서 돌리기)

...

cd progress

python3 backend\_test1.py <http://wlxyzlw.iptime.org:8000/>

python3 backend\_test2.py <http://wlxyzlw.iptime.org:8000/>

...

- \* 프론트엔드 테스트코드 (주의: 시작하기 전에 virtualenv를 켜둘 것)

...

cd progress

python3 myTest.py <http://wlxyzlw.iptime.org:8000/> <http://wlxyzlw.iptime.org:3000/>

...