

Vue PJT 02

1. 목표

- Vue CLI 개발 환경에 대한 이해
- 컴포넌트 기반 구조에 대한 이해
- Vue Router에 대한 이해
- API 서버와의 통신 및 설정에 대한 이해

2. 준비 사항

1. vue

- Node 12.x
- Vue 2.x
- Vue CLI 4.x

2. django

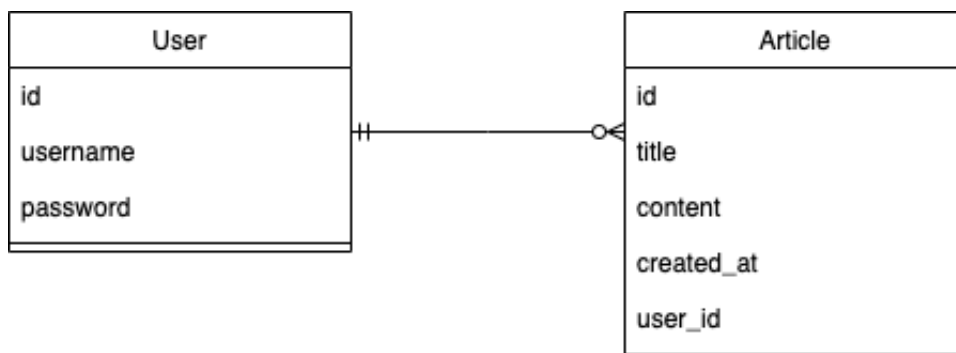
- python 3.6+
- django 2.1.x
- django-rest-framework

3. 기능 명세

3.1. 필수 기능 구현 안내

- 회원 관련 기능
 - 회원가입
 - 로그인
 - 로그인은 토큰 기반 인증으로 진행하며, 쿠키에 저장하여 활용합니다.
- 커뮤니티(게시판) 기능
 - 게시물 생성
 - 게시물 조회(목록/상세보기)
- 아래 예시에 적용된 스타일링은 자유롭게 진행 해주시면 됩니다.

3.2. 데이터베이스 구조



4. 구현 예시

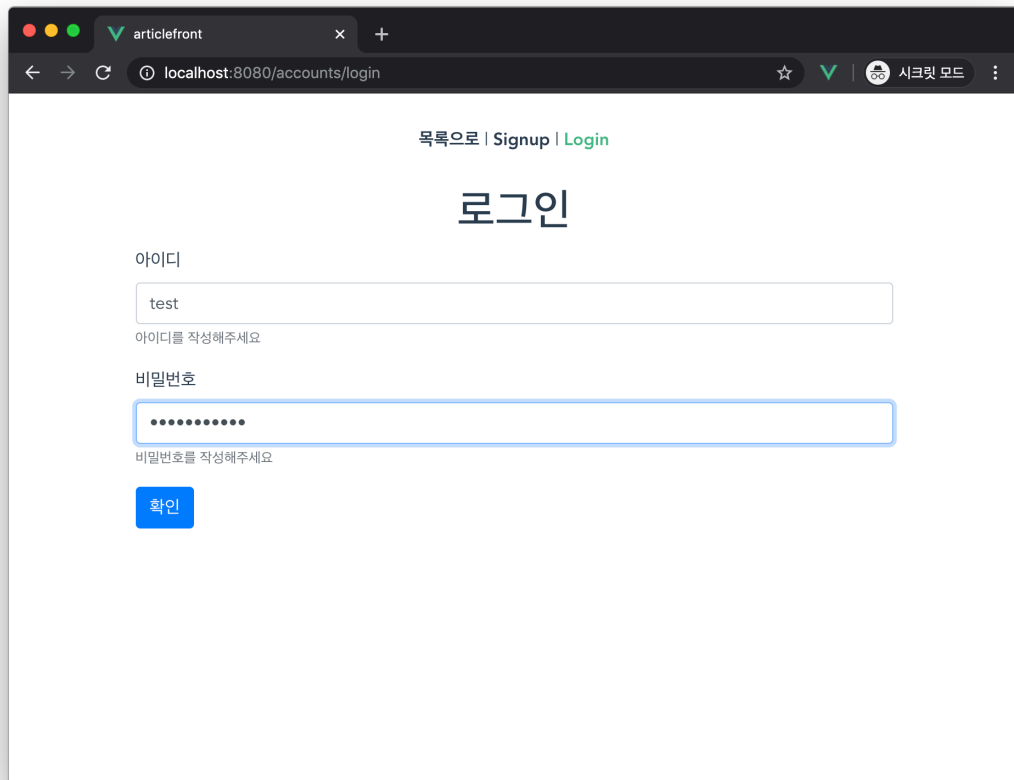
4.1. 컴포넌트별 구현 예시

1. App.vue

- 최상위 컴포넌트로 Navbar와 router-link를 포함한다.
- 비로그인시에는
 - Community, Signup, Login 링크가 노출된다.
- 로그인시에는
 - Community, Logout 링크가 노출된다.

2. LoginView.vue

- 로그인 폼과 함께 로그인 동작을 한다.
- 로그인이 되는 경우 쿠키에 토큰 값이 저장되어야 한다.
- 로그인에 실패하는 경우 에러 메시지를 콘솔에 출력한다.



3. SignupView.vue

- 회원가입 폼과 함께 회원가입 동작을 한다.
- 회원가입에 실패하는 경우 에러 메시지를 콘솔에 출력한다.

articlefront

localhost:8080/accounts/signup

목록으로 | Signup | Login

회원가입

아이디

아이디를 작성해주세요

비밀번호

비밀번호를 작성해주세요

비밀번호 확인

비밀번호를 다시 한번 입력해주세요

확인

4. `ArticleListView.vue`

- 게시글 목록에는 제목과 작성자만 출력된다.

articlefront

localhost:8080/articles/

목록으로 | Logout

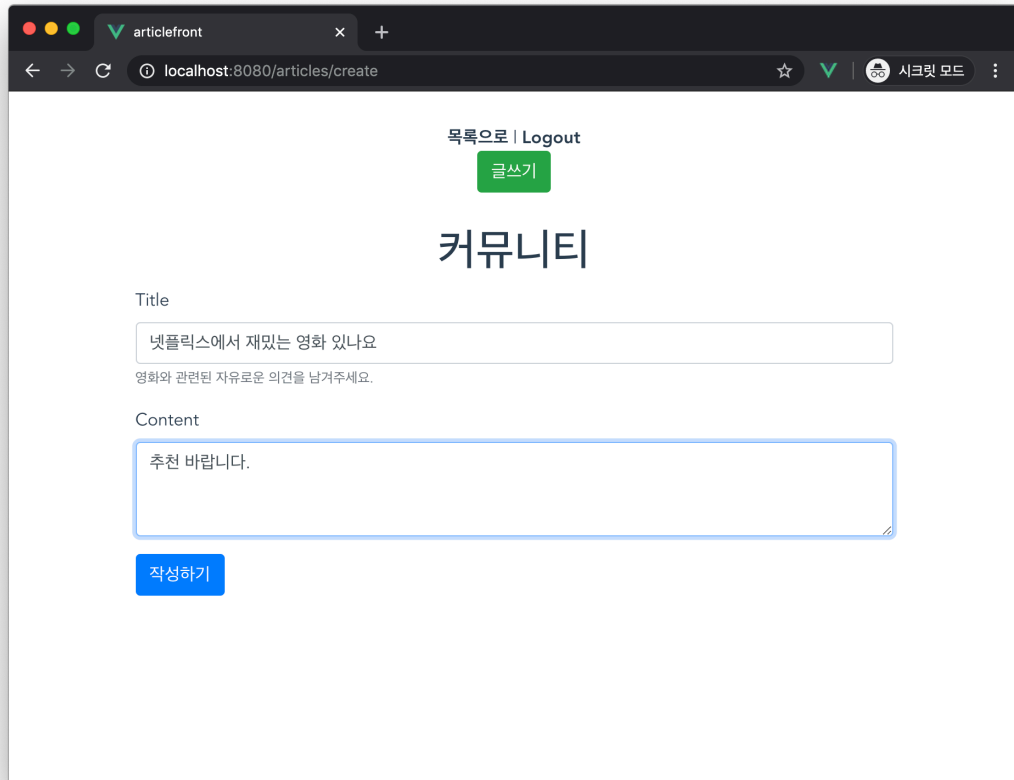
글쓰기

게시판

#	제목	작성자
1	넷플릭스에서 재밌는 영화 있나요	test

5. ArticleCreateView.vue

- 게시글 폼과 함께 제목과 내용을 작성할 수 있다.
- 로그인시에만 작성 가능하다.
- 검증에 실패하는 경우 에러 메시지를 콘솔에 출력한다.



4.2. API 서버 구현 예시

1. 최소 구현 URL

요청메서드	PATH	비고
POST	rest-auth/login/	로그인(header: token, data: username, password)
POST	rest-auth/logout/	로그아웃(header: token)
POST	rest-auth/signup/	회원가입(data : username, password1, password2)
GET	articles/	게시글 목록
POST	articles/create/	게시글 생성(header: token)

5. 제출 안내

제출 기한은 20.06.10(수) 18:00까지 입니다. 제출기한을 엄수하여 주시기 바랍니다.

반드시 README.md 파일에 구현 과정 및 페어 프로그래밍에서 느낀점을 기록하여 제출해야 합니다.

<https://lab.ssafy.com>에 vue_pjt2로 프로젝트를 생성하고 해당 프로젝트에 업로드하여 제출합니다.

각 반 담당 교수님을 Maintainer로 설정해야 합니다.

결과물은 아래와 같은 파일/폴더 구조로 이루어져 있어야 합니다.

제출시 `.gitignore` 를 활용하여, 최소 `venv` 혹은 `node_modules` 폴더가 포함되지 않도록 해주시기 바랍니다.

```
vue_pjt2/  
  README.md  
  movie_api/  
  
  movie_vue/  
    src/  
      components/  
        ...  
      router/  
        index.js  
      views/  
        ...  
    App.vue
```