

Web Studio 2019

4. flask_restful

Contents

1. 유저 리스트에 id 부여하기
2. 글쓰기
3. 댓글 달기
4. 좋아요

유저 리스트에 id 부여하기

id? ?

1. 저번 시간까지는 email을 고유한 값으로 비교했었음
2. 한 데이터(user)가 갖고있는 고유한 값을 id라고 함
3. 일반적으로 1부터 시작하여 증가하는 순으로 부여됨

유저 리스트에 id 부여하기

실습1: 기존 api에 id 부여하기

1. 일단 4_flask_restful_〈여러분영어이름〉 브랜치를 만들고 이동함
: checkout -b 4_flask_restful_〈여러분영어이름〉
2. 4_flask_restful/practice/〈여러분영어이름〉/api.py를 작성 (지난시간에 한거 카피)
3. users.json을 지움
4. post 메소드안에 users.json을 읽어서 현재 유저가 몇명인지 먼저 셈
5. 새로운 유저를 추가할 때, id를 현재 유저수 + 1로 주고 새로 저장함

글쓰기

articles.json

1. id, user_id, title, content를 key로 가지는 리스트를 저장
2. user_id를 이용해 글쓴이를 알 수 있음

id: 1, email: sisobus@vuno.co
id: 2, email: test@vuno.co

id: 1, user_id: 1, title: "A", content: "test"
id: 2, user_id: 1, title: "B", content: "test"
id: 3, user_id: 2, title: "C", content: "test"
id: 4, user_id: 2, title: "D", content: "test"
id: 5, user_id: 2, title: "E", content: "test"

글쓰기

실습2: 글쓰기 구현하기

1. `class ArticleList(Resource)` 클래스를 구현함
2. `def get (self):` 글 리스트를 반환
3. `def post(self):` 글 작성하기, 데이터로는 `user_id`, `title`, `content`를 받음
4. `def put(self):` 글 수정하기, 데이터로는 `article_id`, `title`, `content`를 받음
5. `def delete(self):` 글 삭제하기, 데이터로는 `article_id`를 받음
6. `api.add_resource(ArticleList, '/api/articles')`로 등록함

댓글 달기

comments.json

1. id, user_id, article_id, content를 key로 가지는 리스트를 저장
2. user_id를 이용해 글쓴이를 알 수 있음
3. article_id를 이용해 어떤 글의 댓글인지 알 수 있음

id: 1, email: sisobus@vuno.co
id: 2, email: test@vuno.co

id: 1, user_id: 1, title: "A", content: "test"

id: 2, user_id: 1, title: "B", content: "test"

id: 3, user_id: 2, title: "C", content: "test"

id: 4, user_id: 2, title: "D", content: "test"

id: 5, user_id: 2, title: "E", content: "test"

id: 1, user_id: 1, article_id: 4, ...

id: 2, user_id: 2, article_id: 1, ...

댓글 달기

실습3: 댓글달기 구현하기

1. `class CommentList(Resource)` 클래스를 구현함
2. `def get (self):` 댓글 리스트를 반환
3. `def post(self):` 댓글 작성하기, 데이터로는 `user_id`, `article_id`, `content`를 받음
4. `def put(self):` 댓글 수정하기, 데이터로는 `comment_id`, `content`를 받음
5. `def delete(self):` 댓글 삭제하기, 데이터로는 `comment_id`를 받음
6. `api.add_resource(CommentList, '/api/comments')` 로 등록함

좋아요

아직 미완성 ㅠㅠ

1. 지금 6시 27분 과연 난 늦지 않게 도착할 수 있을 것인가

글에 달려있는 코멘트 API

추가구현: id가 n인 글의 모든 댓글을 가져오는 API 작성

1. `/api/articles/5/comments` : 5번 글의 모든 코멘트를 가져오라는 URI
2. `class ArticleHasCommentList(Resource)` 작성

Q & A

Appendix

Bash 기본 명령어

1. ls : 현재 위치의 파일 리스트
 - ls -al : 숨겨진 파일까지 전부 보기
2. cd : 디렉토리 이동
 - .은 현재 디렉토리, ..은 바로 전 디렉토리를 의미함
 - 예를들어 cd .. 이 명령어는 바로 전 디렉토리로 이동
3. rm : 파일 지우기
 - rm -rf : 디렉토리도 지울 수 있음
4. vi <filename> : vim 에디터를 이용한 파일 작성하기
5. mkdir <directoryname> : 디렉토리(폴더) 만들기
6. pwd : 현재 위치 출력하기

Appendix

Vim

1. 최고의 Text editor
2. 이것만 잘써도 코딩 생산성이 상당히 많이 올라감
3. 대부분의 IDE에는 Vim Plugin이 존재함
4. Vim 쓰세요
5. 두번쓰세요
6. 세번쓰세요
7. 평생쓰세요

Appendix

Vim

1. 명령어 모드와 에디터 모드로 나뉨
2. 맨처음 들어갈 때에는 명령어 모드임
3. 에디터 모드에서 명령어 모드로 바꾸는 것은 esc로 함
4. 명령어 모드에서 에디터 모드로 바꾸는 것은 i, o, a로 함 (i만 알아도 상관 없습니다.)

Appendix

Vim 명령어 모드

1. i는 현재 커서에서 에디터모드로 전환
2. yy는 현재 줄 복사
3. p는 복사한 것을 붙여넣기
4. dd는 현재줄 삭제
5. u는 undo
6. ctrl+r는 redo
7. :wq 는 저장후 종료를 의미함
8. /<search string> 으로 검색할 수 있음
9. :%s/<stringA>/<stringB>/g 는 stringA를 stringB로 대체함
 - :%s/^/₩/₩//g : 맨 앞에 //를 붙임
 - :3,12s/ha/hi/g : 3번줄부터 12번째줄의 ha를 hi로 바꿈