

Spring Project

목차



1. 개요



2. 구성



3. 부족한 점



4. 개선방안



1. 개요



blog.naver.com/ehanuri

- 개발자는 개발 서적이나 리프레쉬를 위한 서적을 읽는 것을 지속해야 됨

- 회원 + 독서 기록을 하면 어떨까?

1. 개요

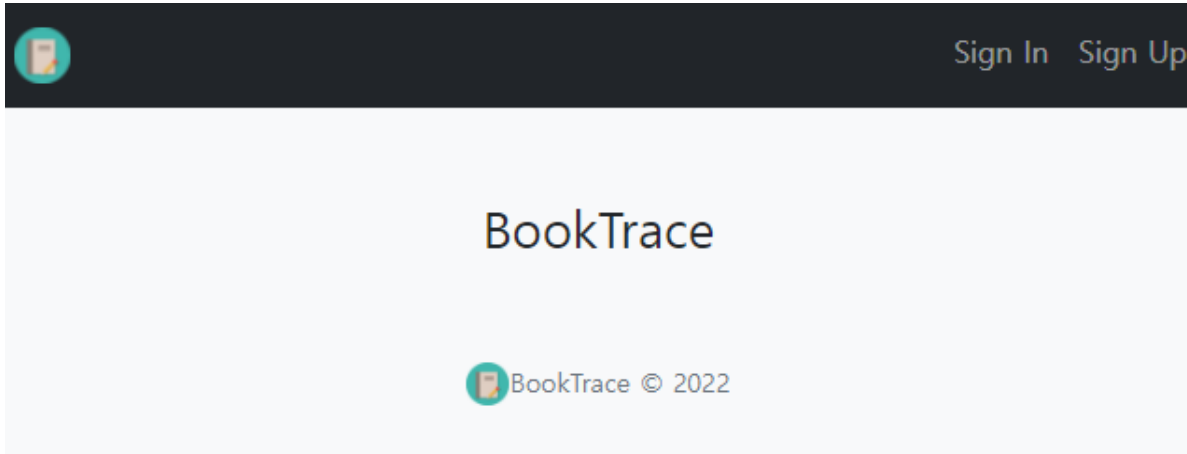


사전조사

- 상용 애플리케이션으로 '북적북적'이 IOS, AOS 모두 지원하는 것으로 확인
- 독서 기록과 캐릭터로 쌓아가는 것을 목표로 하는 것
- 링크([Link](#))

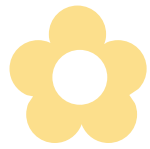


1. 개요

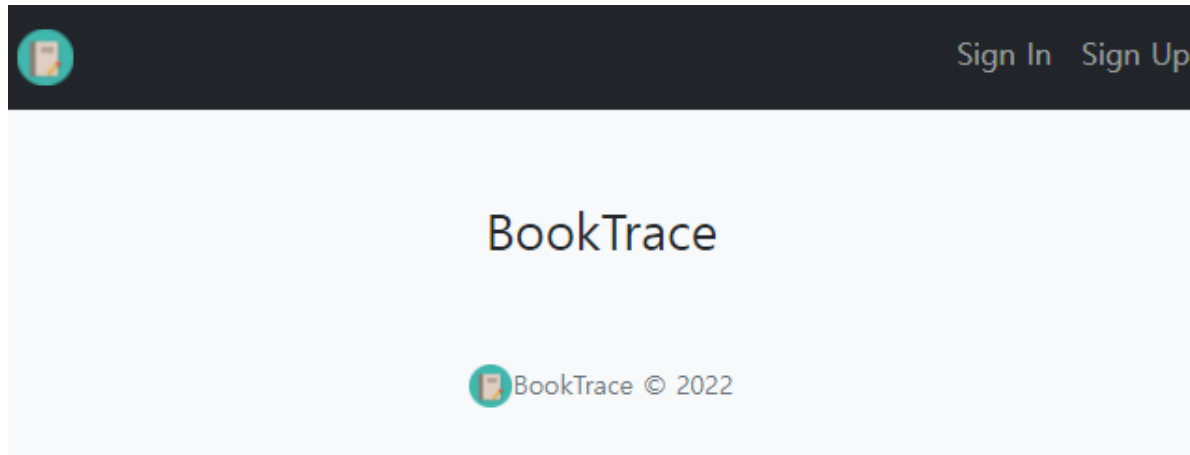


프로젝트 목표

- Spring Boot + Thymeleaf로
SSR(Server Side Rendering)을 활용해서
최대한 회원 기능과 도서 검색을 지원하는 것
을 목표

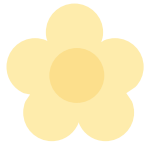


1. 개요



프로젝트 사용 기술

- Language : Java 11, MySQL 8.0.29
H2 1.4.200, BootStrap, JQuery
- Library : SpringBoot 2.6.8, Node 2.2.3, P6spy 1.8.0, Thymeleaf, validation, data-jpa, web, mail, actuator, Lombok, security Cropper, jdenticon 등



2. 구성 - Database

이름 ^			
book			
member			
persistent_logins			

#	이름	데이터 유형	길이/설정
1	book_id	BIGINT	19
2	author	VARCHAR	255
3	book_status	VARCHAR	255
4	description	LONGTEXT	
5	end_date	DATE	
6	expectation	LONGTEXT	
7	isbn	VARCHAR	255
8	publisher	VARCHAR	255
9	read_page	INT	10
10	score	INT	10
11	start_date	DATE	
12	thumbnail	VARCHAR	300
13	title	VARCHAR	255
14	tot_page	INT	10
15	url	VARCHAR	300
16	member_id	BIGINT	19

열: + 추가 - 제거 ▲ 위로 ▼ 아래로			
#	이름	데이터 유형	길이/설정
1	series	VARCHAR	64
2	last_used	DATETIME	6
3	token	VARCHAR	64
4	username	VARCHAR	64

DataBase

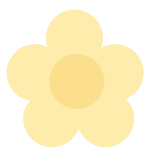
- 간단하게 Book, Member, 로그인 유지

- Book이 Member를 소유하고 있는 단
방향 연관관계를 가집니다.


- 각 테이블의 상세 설명은 생략합니다.

- 로그인 유지 테이블은 이중 토큰 인증
방식입니다.

#	이름	데이터 유형	길이/설정
1	member_id	BIGINT	19
2	created_date	DATETIME	6
3	last_modified_...	DATETIME	6
4	email	VARCHAR	255
5	email_token	VARCHAR	255
6	email_token_g...	DATETIME	6
7	is_valid	BIT	1
8	joined_date	DATETIME	6
9	nickname	VARCHAR	255
10	password	VARCHAR	255
11	bio	VARCHAR	255
12	company	VARCHAR	255
13	favorite	VARCHAR	255
14	image	LONGTEXT	
15	job	VARCHAR	255
16	url	VARCHAR	255



2. 구성 - 회원

 Sign In Sign Up

회원 가입

닉네임

공백없이 문자와 숫자로만 3자 이상 20자 이내로 입력하세요. 가입후에 변경할 수 있습니다.


이메일

BookTrace는 사용자의 이메일을 공개하지 않습니다.

비밀번호

8자 이상 50자 이내로 입력하세요. 영문자, 숫자, 특수기호를 사용할 수 있으며 공백은 사용할 수 없습니다.

☐ 약관 에 동의하시면 가입하기 버튼을 클릭하세요.

 BookTrace © 2022

```
@Email
@NotBlank
private String email;

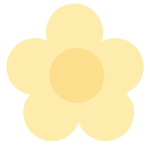
@NotBlank
@Length(min = 8, max = 50)
private String password;

@NotBlank
@Length(min = 3, max = 20)
@Pattern(regexp = "[ㄱ-ㅎ가-힣a-zA-Z\\d_-]{3,20}$")
private String nickname;
```

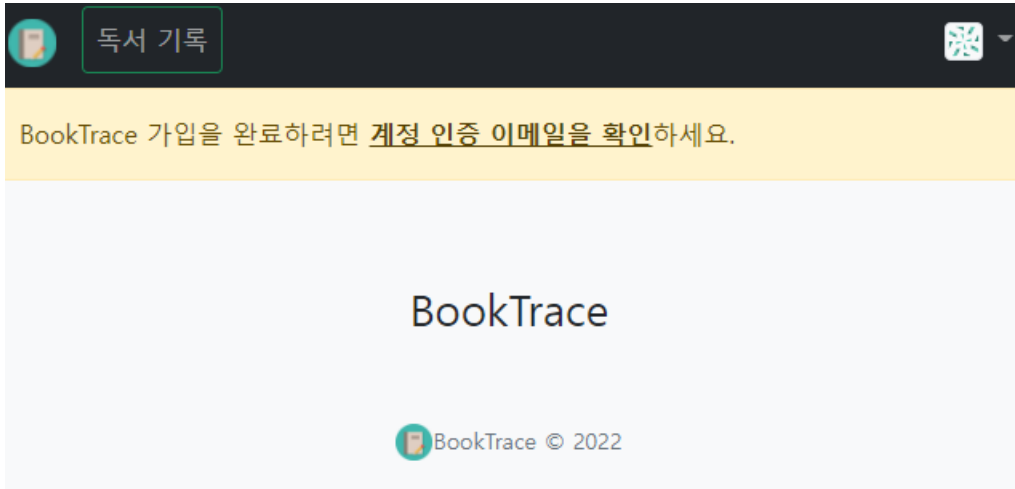
회원 가입

- Form을 이용해서 Post 전송을 사용합니다.
- 닉네임, 이메일, 비밀번호로만 간편하게 가입이 가능합니다.
- 가입시 이메일 인증을 요구합니다.
- 가입 시 Validation을 서버 단에서 Entity가 아닌 DTO(Data Transform Object)에서 검증합니다.

mem...	created_date	last_modified_date	email	email_token	email_token_gen...	is_valid	joined_date	nickname	password	bio	compan
1	2022-06-01 ...	2022-06-01 04:37:...	rylah@kakao.com	026f807d-606...	2022-06-01 04:3...	0	(NULL)	Rylah	{bcrypt}\$2...	(NULL)	(NULL)

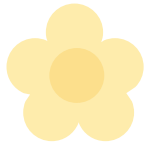


2. 구성 - 회원

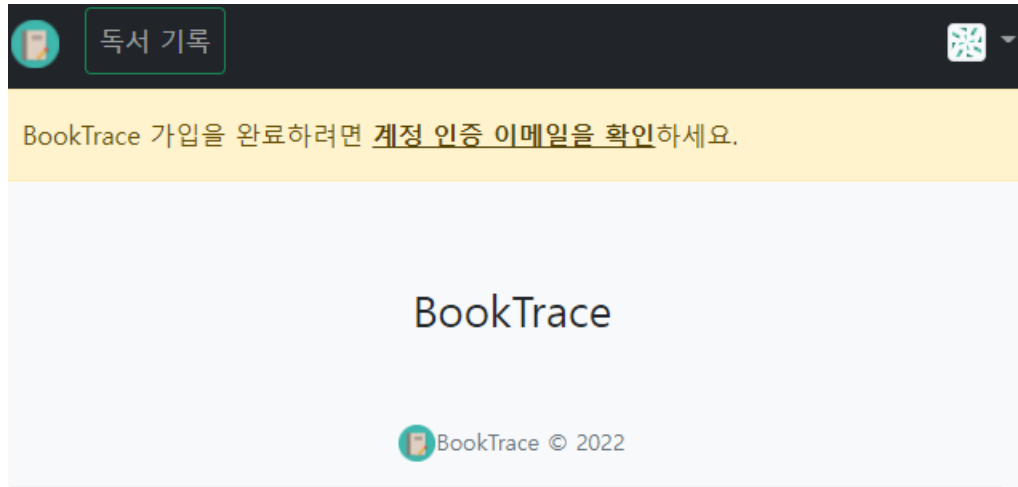


이메일 인증

- SMTP 프로토콜을 이용해서 이메일 인증을 보낼 수 있습니다.
- 원래라면 직접 서버단에서 해야 되지만 gmail을 이용했습니다.
- 실무에서는 사용이 불가능합니다.



2. 구성 - 회원



dev.booktrace...

BookTrace 회원 가입 인증 링크

BookTrace 회원 가입 인증 링크

보낸사람 <dev.booktrace@gmail.com> 주소록 추가 | 수신차단 | 관심친구

받는사람 <rylah@kakao.com> 주소록 추가

안녕하세요, Rylah님!

BookTrace 가입 인증을 위해 링크를 클릭하세요

[이메일 인증](#)

링크가 동작하지 않는 경우에는 아래 URL을 웹브라우저에 복사해서 붙여 넣으세요.

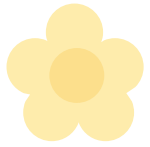
이메일 인증

- SMTP 프로토콜을 이용해서 이메일 인증을 보낼 수 있습니다.

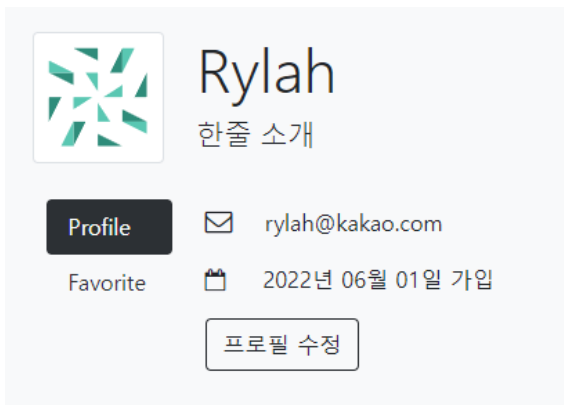
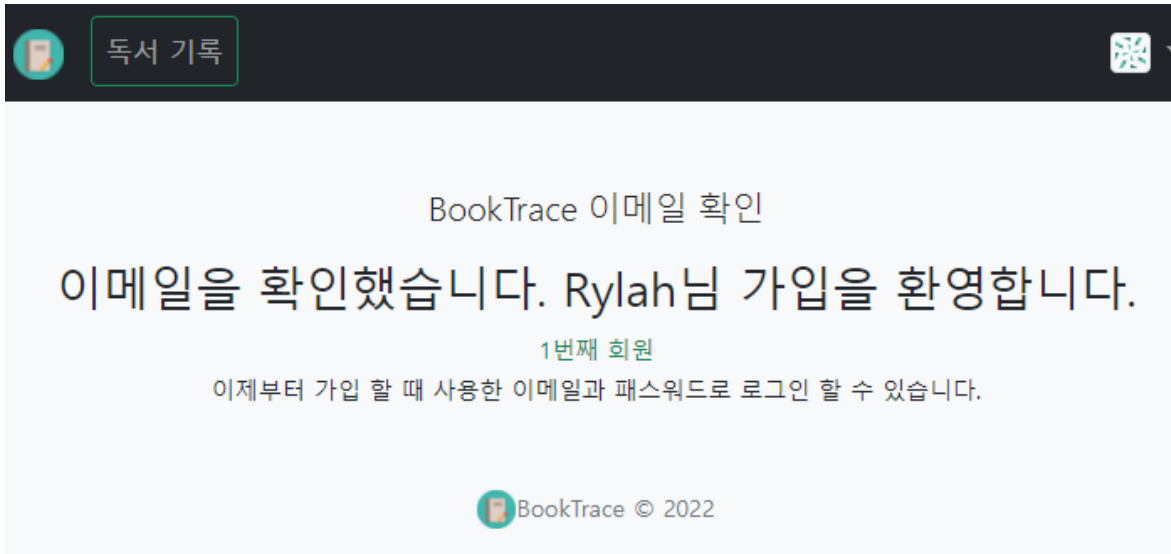
- 원래라면 직접 서버단에서 해야 되지만 gmail을 이용했습니다.

- 실무에서는 사용이 불가능합니다.

- 인증 메일 확인입니다.



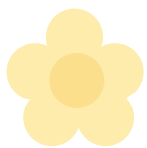
2. 구성 - 회원



is_valid	joined_date	nickname
1	2022-06-01...	Rylah

이메일 인증

- SMTP 프로토콜을 이용해서 이메일 인증을 보낼 수 있습니다.
- 원래라면 직접 서버단에서 해야 되지만 gmail을 이용했습니다.
- 실무에서는 사용이 불가능합니다.
- DB에서 인증된 것과 인증 날짜가 기록됩니다.



2. 구성 - 로그인

BookTrace

Login

Email or NickName

Email@email.com

가입할 때 사용한 이메일 또는 닉네임을 입력하세요.

비밀번호

비밀번호가 기억나지 않으시나요? [이메일로 로그인하기](#)

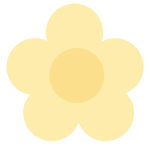
☒ 로그인 유지

[Sign In](#) 아직 회원이 아니신가요? [Sign Up](#)

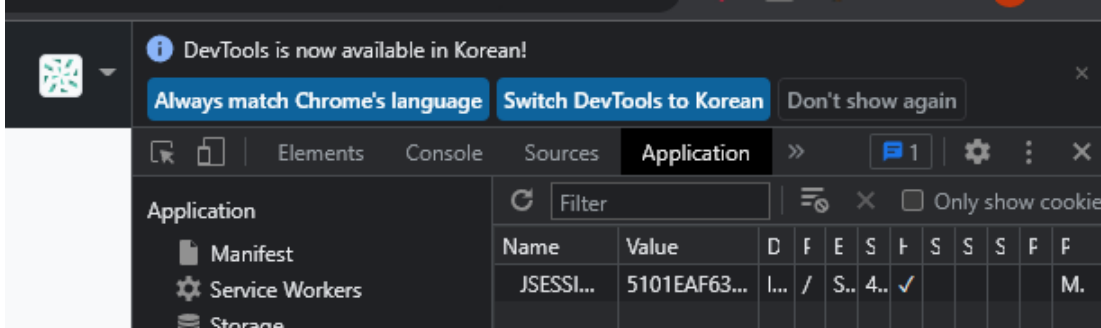
로그인

- 로그인에는 다양한 방식을 사용할 수 있지만, 가장 기본적인 쿠키-세션 2중 방식을 채택했습니다.

- RESTAPI, OAuth2.0 등을 사용한다면 이 방식 보다는 Jwt와 같은 방식이 더 낫다고 판단됩니다.

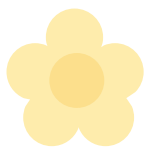


2. 구성 - 로그인

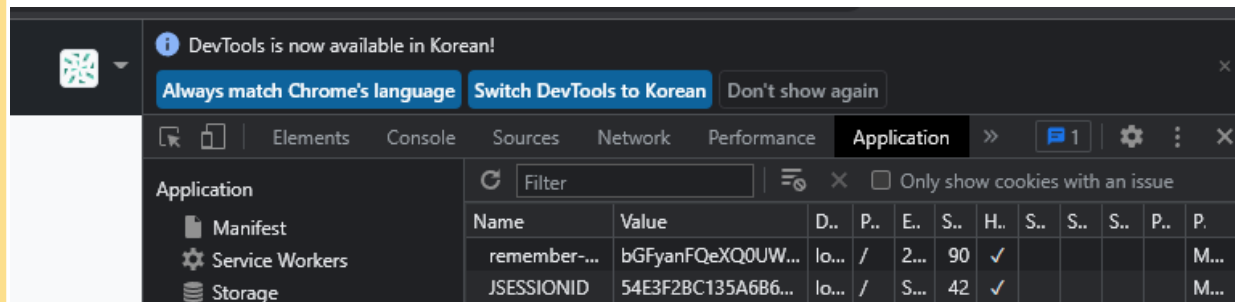


로그인 유지 미 선택

- 로그인 이후 서버에서 JSESSIONID를 발급하고 클라이언트는 이 정보를 COOKIE에 저장합니다.
- 클라이언트는 JSESSION을 기반으로 로그인이 되어있는지를 판별합니다.
- 서버를 켜다 켜게 되면 관리하는 세션이 없어집니다.
- Tomcat 기본 설정인 30분 유지에 제한이 걸립니다.



2. 구성 - 로그인

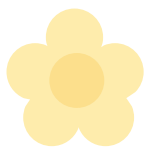


로그인 유지 선택

- 쿠키를 하나 더 사용 (인증 정보 저장)
- > 악의적인 쿠키 탈취의 위험성

조금 개선한 방법

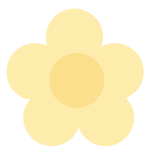
- 랜덤 토큰 값과 랜덤 시리즈 값을 사용
- 시리즈는 발급 이후 고정된 값
- 쿠키가 탈취 당했다면 탈취 한쪽에서는 유효한 값을 사용하고 탈취 당한쪽에서 유효하지 않은 값을 사용하면 모두 무효화
- Spring Security에서 제공



2. 구성 - 로그인

```
public class SecurityConfig extends WebSecurityConfigurerAdapter {  
  
    @Override  
    protected void configure(HttpSecurity http) throws Exception {  
        http.rememberMe() // userDetailsService와 token을 관리할 repository를 설정하고 기억해준다.  
            .userDetailsService(memberService)  
            .tokenRepository(tokenRepository());  
    }  
}
```

```
@Table(name = "persistent_logins")  
@Entity  
@Getter  
public class PersistentLogins {  
    @Id  
    @Column(length = 64)  
    private String series;  
  
    @Column(length = 64)  
    private String username;  
  
    @Column(length = 64)  
    private String token;  
  
    @Column(name = "last_used", length = 64)  
    private LocalDateTime lastUsed;  
}
```



2. 구성 - 로그인

BookTrace

Login

Email or NickName

Email@email.com

가입할 때 사용한 이메일 또는 닉네임을 입력하세요.

비밀번호

비밀번호가 기억나지 않으시나요? [이메일로 로그인하기](#)

☒ 로그인 유지

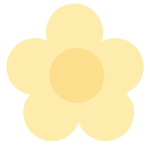
아직 회원이 아니신가요? [Sign Up](#)

로그인 유지 선택 시

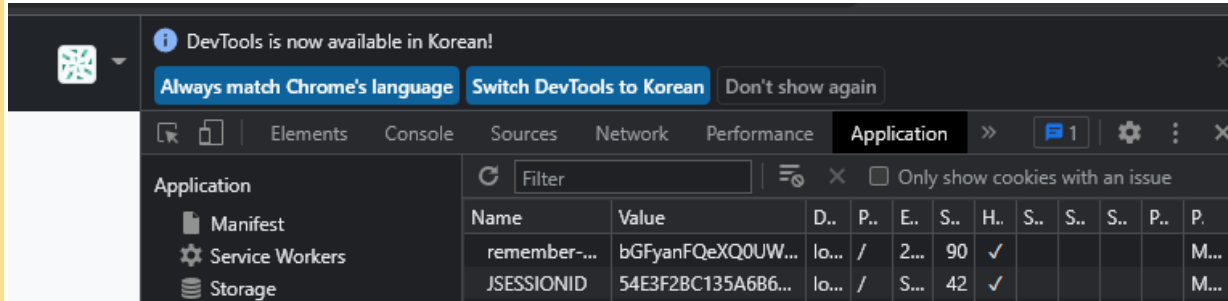
- 그래서 로그인 유지를 선택하면

Persisitent_login table에 인증하는 시리즈와 토큰이 저장되게 된다. 다음과 같다

series	last_used	token	username
larjqPyt4QfPOvLHjp8JVg...	2022-06-01 04:54:49.435000	f3sDs0Pp/VPE6sc3xJ57...	Rylah

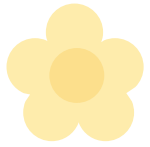


2. 구성 - 로그인

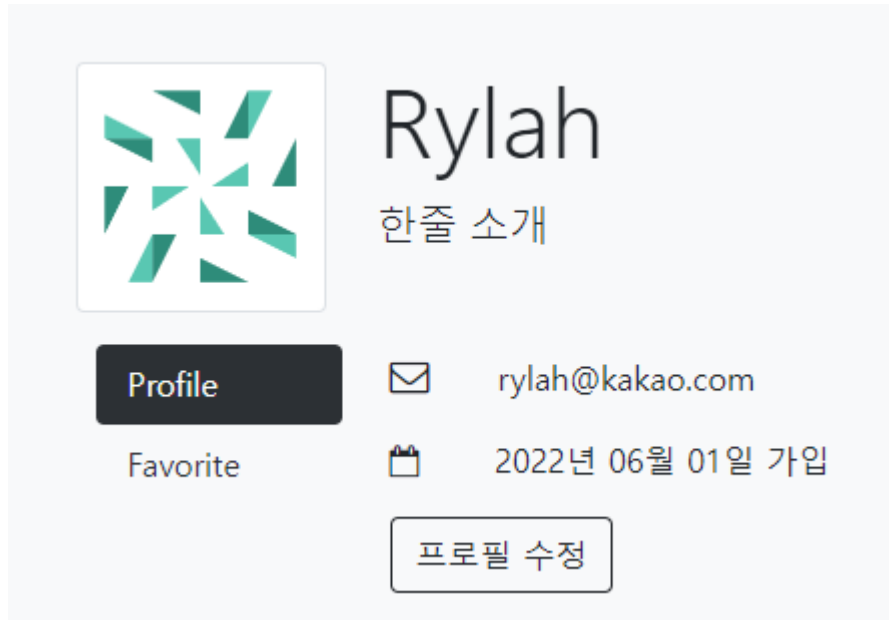


로그인 유지 선택

- 다시 돌아와서 RememberMe 토큰이 이 시리즈의 역할을 수행해주는 것입니다.
- 이 토큰을 삭제하게 되면 추후 서버 재시작 혹은 브라우저 재 시작시에 로그인 유지가 풀립니다.

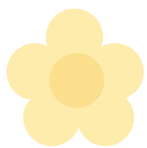


2. 구성 - 프로필



프로필

- 프로필 기본 이미지는 Jdenticon Library를 사용해서 문자열에 따른 이미지 생성을 이용
- 가입 일자는 이메일 인증 날짜와 동일
- 프로필 수정을 하면 추가 프로필이 나타나는 방식



2. 구성 - 프로필

프로필

비밀번호

관심태그

계정 관리

Rylah

한 줄 소개
한줄로 자신을 표현해보세요.
40자 이내로 입력하세요.
링크

본인을 소개하는 링크(Blog, Github)를 추가하세요.
직업

ex) 개발자, 취업준비생, 군인, 공무원, 무직 등
관심 분야

관심 분야를 입력해주세요.

수정하기

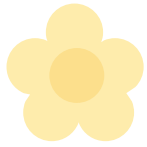
프로필 이미지

파일 선택

선택된 파일 없음

프로필 수정

- 기본적인 한줄 자기 소개, 링크(블로그 홈페이지 등), 직업, 관심 분야를 설정할 수 있습니다.
- 프로필 이미지는 Cropper 라이브러리로 설정할 수 있습니다.
- 예시는 다음 화면으로..



2. 구성 - 프로필

프로필

비밀번호

관심태그

계정 관리

Rylah

한 줄 소개

안녕하세요 Rylah 입니다.

40자 이내로 입력하세요.

링크

http://github.com/rylahs

본인을 소개하는 링크(Blog, Github)를 추가하세요.

직업

개발자 지망생

ex) 개발자, 취업준비생, 군인, 공무원, 무직 등

관심 분야

프로그래밍, 게임

관심 분야를 입력해주세요.

수정하기

프로필 이미지

파일 선택

test.jpg

자르기

확인

취소

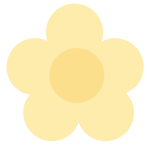
프로필 수정

- 자기소개를 모두 입력

- Cropper 라이브러리를 이용해 이미지를 불러오고 원하는 영역을 잘라서 아래쪽에 결과를 확인할 수 있습니다.

- 이를 저장하면 Database의 Member에 Profile쪽에 저장이 됩니다.

- SingleTable 전략이라 Member 테이블에 그대로 데이터가 들어갑니다.



2. 구성 - 프로필

bio	company	favorite	image	job	url
안녕하세요 Rylah 입니다.	(NULL)	프로그래밍, 게임	data:image/png;base64,iVB...	개발자 지망생	http://github.com/rylahs

프로필을 수정하였습니다.

Rylah

한 줄 소개

안녕하세요 Rylah 입니다.

40자 이내로 입력하세요.

링크

http://github.com/rylahs

본인을 소개하는 링크(Blog, Github)를 추가하세요.

직업

개발자 지망생

ex) 개발자, 취업준비생, 군인, 공무원, 무직 등

관심 분야

프로그래밍, 게임

관심 분야를 입력해주세요.

수정하기

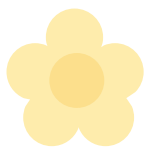
프로필 이미지



파일 선택 선택된...일 없음

프로필 수정

- navbar와 프로필 모두 수정이 된 것을 확인
- DB에도 수정이 된 것을 확인



2. 구성 - 비밀번호 변경

프로필

비밀번호

관심태그

계정 관리

비밀번호 변경

새 비밀번호

새 비밀번호를 입력하세요.
비밀번호를 입력하세요.

새 비밀번호 확인

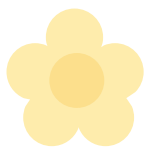
새 비밀번호를 다시 한번 입력하세요.
비밀번호를 다시 입력하세요.

비밀번호 변경

비밀번호 변경

- DTO를 이용해서 비밀번호를 Valid 검증

- 두 비밀번호가 양식과 맞으며 일치하면 비밀번호 변경



2. 구성 - 비밀번호 변경

프로필

비밀번호

관심태그

계정 관리

비밀번호 변경

새 비밀번호

새 비밀번호를 입력하세요.
비밀번호를 입력하세요.

새 비밀번호 확인

새 비밀번호를 다시 한번 입력하세요.
비밀번호를 다시 입력하세요.

비밀번호 변경

비밀번호 변경

- DTO를 이용해서 비밀번호를 Valid 검증

- 두 비밀번호가 양식과 맞으며 일치하면 비밀번호 변경

비밀번호를 변경했습니다. X

비밀번호 변경

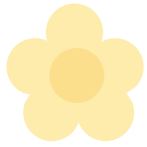
새 비밀번호

새 비밀번호를 입력하세요.

새 비밀번호 확인

새 비밀번호를 다시 한번 입력하세요.

비밀번호 변경



2. 구성 - 닉네임 변경

닉네임 변경

⚠ 닉네임을 변경하시면 프로필 페이지 링크도 변경됩니다.

Rylah

공백 없이 문자와 숫자로만 3자 이상 20자 이내로 입력하세요.

변경

✔ 닉네임이 수정되었습니다.



닉네임 변경

⚠ 닉네임을 변경하시면 프로필 페이지 링크도 변경됩니다.

Lylah

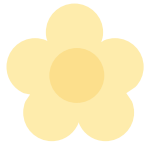
공백 없이 문자와 숫자로만 3자 이상 20자 이내로 입력하세요.

변경

닉네임 변경

- DTO를 이용해서 닉네임 Valid 검증

- 로그인 된 상태에서만 접근 가능(Spring Security 예외에 포함되지 않음)



2. 구성 - 테스트

✓ <default package>	2 sec 578 ms
✓ BooktraceApplicationTests	79 ms
✓ contextLoads()	79 ms
✓ MemberControllerTest	818 ms
✓ 회원 가입 화면 진입 성공 테스트	303 ms
✓ 회원가입 처리 : 입력값 오류3 (이메일 오류)	146 ms
✓ 회원가입 처리 : 입력값 오류1 (닉네임 길이 오류)	21 ms
✓ 회원가입 처리 : 입력값 오류2 (닉네임 정규식 오류)	11 ms
✓ 회원가입 처리 : 성공	216 ms
✓ 회원 가입 성공 후 비밀번호 검증	65 ms
✓ 회원가입 처리 : 입력값 오류4 (비밀번호 오류)	9 ms
✓ 인증 메일 확인: 잘못된 링크 상황	21 ms
✓ 인증 메일 확인: 유효한 링크 상황	26 ms
✓ MainControllerTest	887 ms
✓ 이메일로 로그인 실패하기	216 ms
✓ 로그아웃_성공	63 ms
✓ 이메일로 로그인 성공하기	117 ms
✓ 닉네임으로 로그인 성공하기	116 ms
✓ 이메일로 로그인 실패하기	117 ms
✓ 닉네임으로 로그인 실패하기	121 ms
✓ 닉네임으로 로그인 실패하기	137 ms
✓ SettingsControllerTest	794 ms
✓ 프로필 수정 성공	77 ms
✓ 프로필 조회 : 성공	102 ms
✓ 비밀번호 변경 폼 진입 테스트	77 ms
✓ 비밀번호 변경 실패(길이가 김)	126 ms
✓ 비밀번호 변경 실패(길이가 짧음)	121 ms
✓ 비밀번호 변경 실패(일치하지 않음)	121 ms
✓ 비밀번호 변경 성공	170 ms

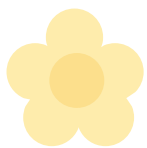
JUnit5 테스트

- JUnit5로 각 기능별로 성공 실패 케이스들을 구현

```
import static org.assertj.core.api.Assertions.assertThat;
```

- 회원 가입, 로그인, 프로필 관련 단위 테스트 작성

- 테스트를 겸해서 개발하면 테스트가 깨지는 지 여부를 통해 기능을 검증하기 편하다

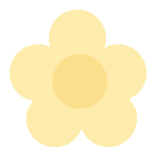


2. 구성 - 테스트

```
@Test
@DisplayName("회원 가입 화면 진입 성공 테스트")
public void signUpFormAccess() throws Exception {
    mockMvc.perform(MockMvcRequestBuilders.get( urlTemplate: "/sign-up"))
        .andExpect(MockMvcResultMatchers.status().isOk()) // HTTP Status
        .andExpect(MockMvcResultMatchers.view().name( expectedViewName: "member/sign-up"))
        .andExpect(MockMvcResultMatchers.model().attributeExists( ...name
        .andExpect(unauthenticated());
}

@Test
@DisplayName("회원가입 처리 : 입력값 오류1 (닉네임 길이 오류)")
public void signUpSubmitWithNickNameValidLength() throws Exception {
    mockMvc.perform(post( urlTemplate: "/sign-up")
        .param( name: "nickname", ...values: "Bo")
        .param( name: "email", ...values: "booktrace@admin.com")
        .param( name: "password", ...values: "1234!!!!")
        .with(csrf()))
        .andExpect(status().isOk())
        .andExpect(view().name( expectedViewName: "member/sign-up"))
        .andExpect(unauthenticated());
}
```

JUnit5 테스트 - 테스트 예시



2. 구성 - 도서 검색(미완성)


BookTrace

BookSearch

검색어

검색어를 입력하세요.

검색

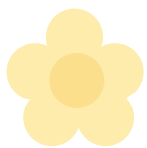
 BookTrace © 2022

도서 검색

- Naver, Kakao, Aladin API를 통한 검증 완료

- 책 제목을 검색하면 그에 따른 결과를 JSON으로 반환

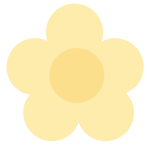
```
{
  documents: [
    {
      authors: [
        "박상현"
      ],
      contents: "뇌를 자극하여 알고리즘을 학습하다! 프로그래머 박상현의 『뇌를 자극하는 알고리즘』, 검증된 코드를 제공하여 직접 분석하고 응용할 수 있도록 돕고 있다. 이 책은 알고리즘",
      datetime: "2009-09-15T00:00:00.000+09:00",
      isbn: "8979146876 9788979146875",
      price: 25000,
      publisher: "한빛미디어",
      sale_price: 22500,
      status: "정상판매",
      thumbnail: "https://search1.kakaocdn.net/thumb/R120x174.q85/?fname=http%3A%2F%2F",
      title: "알고리즘(뇌를 자극하는)",
      translators: [ ],
      url: "https://search.daum.net/search?w=bookpage&bookid=1104126&q=%EC%95%8C%",
    }
  ],
}
```



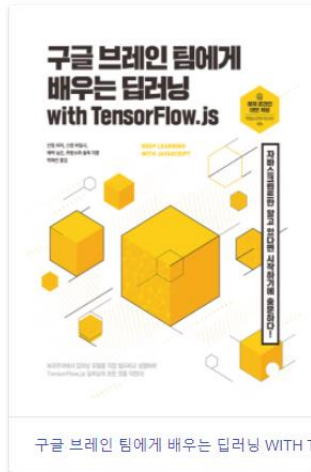
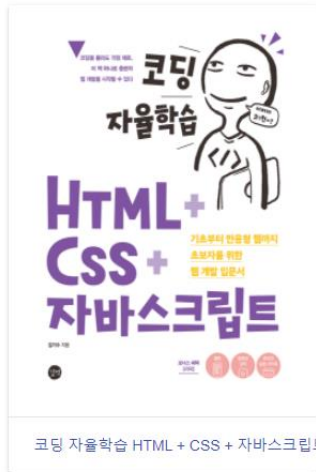
3. 부족한 점

왜 백엔드와 프론트엔드로 분리되었는가?

- JSP, Thymeleaf 등 템플릿 엔진으로 SSR이 불가능하지는 않지만 개발 속도가 현저하게 느리다는 단점이 존재
- REST API와 그에 맞는 프론트엔드 (Vue, React)로 수행하는게 더 효율적이고, 더 빠르고 많이 사용자에게 서비스를 제공할 수 있다는 것을 알게 되었습니다.
- Jwt나 OAuth2.0을 도입하지 못한 것, 요즘은 소셜 로그인이 대세인데 예전 방식의 로그인을 만들어보고 싶었습니다.



4. 개선 방안



REST API로 개선 및 React 적용
- Template Engine의 Thymeleaf가 그나마 편리하다는 것은 확인했습니다.

- 하지만 REST API를 설계하고 응답을 확인한 다음 프론트에서 해결하는 것이 훨씬 더 생산성 있는 방식이라는 것을 절실히 느꼈습니다.

- 옆의 그림과 같은 방식의 결과를 제공

Q&A

