

SHOUT · OUR · PASSION · TOGETHER

ODo IT SOPT O

7 차 세 미 나

SHOUT OUR PASSION TOGETHER
SOPT

모든 설명은 Window 10 설명 기준입니다.

01

정정 & 보충

1. Request Header
2. @RequestPart
3. Jwt 만료 기한
4. port 변경

02

클라이언트 합동 세미나

1. DB
2. 기능
3. 조 편성

○

01

○

정정 & 보충



01



Request Header

Request Header

1. 요청 헤더
2. Content-Type : 클라이언트가 서버에게 Request 할 때 같이 보내주는 콘텐츠 타입 키 값
 1. 파일을 같이 전송 받는 API라면 Request Header의 Content-Type 키 값에 Multipart/form-data을 넣어서 요청해야 한다.
 2. 파일을 전송 받는 API가 아니라면 Content-Type 키 값에 Application/json을 넣어서 요청해야 한다.
 3. Controller의 @RequestBody는 요청 데이터 타입이 Application/json인 것만 받기 때문이다.
3. Authorization : 클라이언트가 서버에게 Request 할 때 자신의 자격을 보내주는 키 값
 1. 해당 API가 토큰(인증)이 필요한 API라면 Request Header의 Authorization 키 값에 토큰 값을 넣어서 요청해야 한다.

Request Header 예시 - login

로그인

| 메소드 | 경로 | 짧은 설명 |
|------|--------|-------|
| POST | /login | 로그인 |

요청 헤더

```
Content-Type: application/json
```

```
@PostMapping("/login")  
public ResponseEntity login(@RequestBody final SignUpReq signUpReq)
```

1. 로그인은 파일 업로드 기능이 있는 API가 아니기 때문에 Content-Type이 application/json이다.
2. @RequestBody Annotation으로 데이터를 받으면 Content-Type은 application/json이다.

Request Header 예시 - 회원 가입

회원 가입

| 메소드 | 경로 | 짧은 설명 |
|------|--------|-------|
| POST | /users | 회원 가입 |

요청 헤더

```
Content-Type: multipart/form-data
```

```
@PostMapping("/users")  
public ResponseEntity saveUser(final SignUpReq signUpReq)
```

1. 여기서 회원 가입 API는 사진을 같이 전송 받는 기능이 있는 API이기 때문에 Content-Type이 multipart/form-data이다.
2. @RequestBody Annotation 없이 데이터를 받으면 Content-Type은 multipart/form-data 혹은 application/x-www-form-urlencoded이다.
3. 파일이 있으면 multipart/form-data 없으면 application/x-www-form-urlencoded이다.

Request Header 예시 - 회원 정보 수정

회원 정보 수정

| 메소드 | 경로 | 짧은 설명 |
|-----|-----------------|----------|
| PUT | /users/{userId} | 회원 정보 수정 |

요청 헤더

```
Content-Type: multipart/form-data
Authorization: token
```

1. 여기서 회원 정보 수정 API는 사진을 같이 전송 받는 기능이 있는 API이기 때문에 Content-Type이 multipart/form-data 이다.
2. 회원 정보 수정은 인증이 필요한 API 이므로 Authorization에 token을 넣어서 보내줘야 한다.
3. 토큰 데이터를 뽑아 내려면 @RequestHeader Annotation을 사용하면 된다.

```
@Auth
@PutMapping("/users/{userId}")
public ResponseEntity updateUser(
    @RequestHeader(value = "Authorization") final String header,
    @PathVariable("userId") final int userId,
    final SignUpReq signUpReq) {
```


Request Header 예시 - 회원 탈퇴

회원 탈퇴

| 메소드 | 경로 | 짧은 설명 |
|--------|-----------------|-------|
| DELETE | /users/{userId} | 회원 탈퇴 |

요청 헤더

```
Content-Type: application/json  
Authorization: token
```

```
@Auth  
@DeleteMapping("/users/{userId}")  
public ResponseEntity deleteUser(  
    @RequestHeader(value = "Authorization") final String header,  
    @PathVariable("userId") final int userId) {
```

1. 회원 탈퇴는 파일 업로드 기능이 있는 API가 아니기 때문에 Content-Type이 application/json이다.
2. 인증이 필요한 API이므로 Authorization에 token을 받아서 검증해야 한다.



01



@RequestPart

@RequestParam

```
public ResponseEntity signup(  
    SignUpReq signUpReq,  
    @RequestParam(value = "profile", required = false) final MultipartFile profile  
) {
```

5차 세미나 내용. 변경 전

```
@PostMapping("/users")  
public ResponseEntity saveUser(final SignUpReq signUpReq) {
```

```
public class SignUpReq {  
  
    private String name;  
    private String email;  
    private String password;  
    private String profileUrl;  
    private MultipartFile profile;
```

변경 후

@RequestParam

1. @RequestParam annotation을 이용해 MultipartFile을 받지 않아도, 전송 받을 객체 안에 MultipartFile의 필드가 있다면 자동으로 받아진다. (Default 생성자와 Setter 메소드를 통해서 자동으로 받아진다.)
2. 만약 파일을 배열(여러 개)로 받고 싶다면 MultipartFile[] profile; 로 변경하면 된다. 이떄 키 값이 항상 같아야 배열로 받아진다. - 2차 세미나 참조

```
public class SignUpReq {  
  
    private String name;  
    private String email;  
    private String password;  
    private String profileUrl;  
    private MultipartFile profile;  
}
```

```
public class ContentReq {  
  
    private int contentIdx;  
    private int userIdx;  
    private String body;  
    private MultipartFile[] photo;  
}
```

파일 1개 받을 때와 여러 개 받을 때



01



Jwt 만료 기한

Jwt 만료 기한 설정

1. 현재 JwtService를 통해 생성된 token엔 만료 기한 설정이 없어서 같은 사람이 로그인 할 경우 token이 항상 같다.
2. 만료 기한을 설정하게 된다면 시간 값은 매번 다르기 때문에 같은 사람이 로그인 해도 token이 다르다.

```
public String create(final int user_idx) {  
    try {  
        //토큰 생성 빌더 객체 생성  
        JWTCreator.Builder b = JWT.create();  
        //토큰 생성자 명시  
        b.withIssuer(ISSUER);  
        //토큰 payload 작성, key - value 형식, 객체도 가능  
        b.withClaim("user_idx", user_idx);  
        //만료 날짜 지정, 1달  
        b.withExpiresAt(expiresAt());  
        //토큰 해싱해서 반환  
        return b.sign(Algorithm.HMAC256(SECRET));  
    } catch (JWTCreationException JwtCreationException) {  
        Log.info(JwtCreationException.getMessage());  
    }  
    return null;  
}
```

```
private Date expiresAt() {  
    Calendar cal = Calendar.getInstance();  
    cal.setTime(new Date());  
    //한달 24*31  
    cal.add(Calendar.HOUR, 744);  
    return cal.getTime();  
}
```

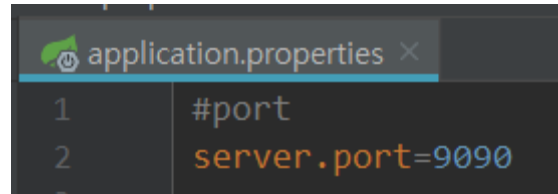
○

01

○

port 변경

port 변경



1. 내장 Tomcat 때문에 Spring-Boot는 기본 port 번호가 8080이다.
2. application.properties에 port 번호를 명시함으로써 변경할 수 있다.



01



앱잼 당부사항

당부 사항

1. A시드는 팀의 모든 것을 책임지는 자리가 아닙니다. 절대 부담가지지 마세요.
2. 모든 팀에 적어도 한명은 프로젝트 경험이 있는 사람을 배정하는 것이 프로젝트 진행에 도움이 되기 때문에 A시드를 선정한 것입니다.
3. A시드가 아닌 사람은 무조건 A시드에게 의존하지 말고 스스로 하는데 까지 노력 하는 것이 자신의 실력 향상에 도움이 됩니다. 그리고 이것을 통해 발전하는 것이 앱잼의 궁극적인 목표입니다.
4. A시드 역시 자신이 모든 것을 다 하려고 하면 안됩니다. 다른 팀원과 같이 개발을 해야 서로 win-win할 수 있습니다.
5. 앱잼이 망해도 누가 뭐라고 A시드를 탓하지 않습니다. 절대 부담가지지 마세요.
6. A시드도 사람입니다. 힘들면 파트장을 부르세요.
7. 모든 사람이 중간 이탈 없이 앱잼을 마무리 했으면 좋겠습니다. (탈주는 안돼요 ㅠ)

○

02

○

클라이언트 합동 세미나

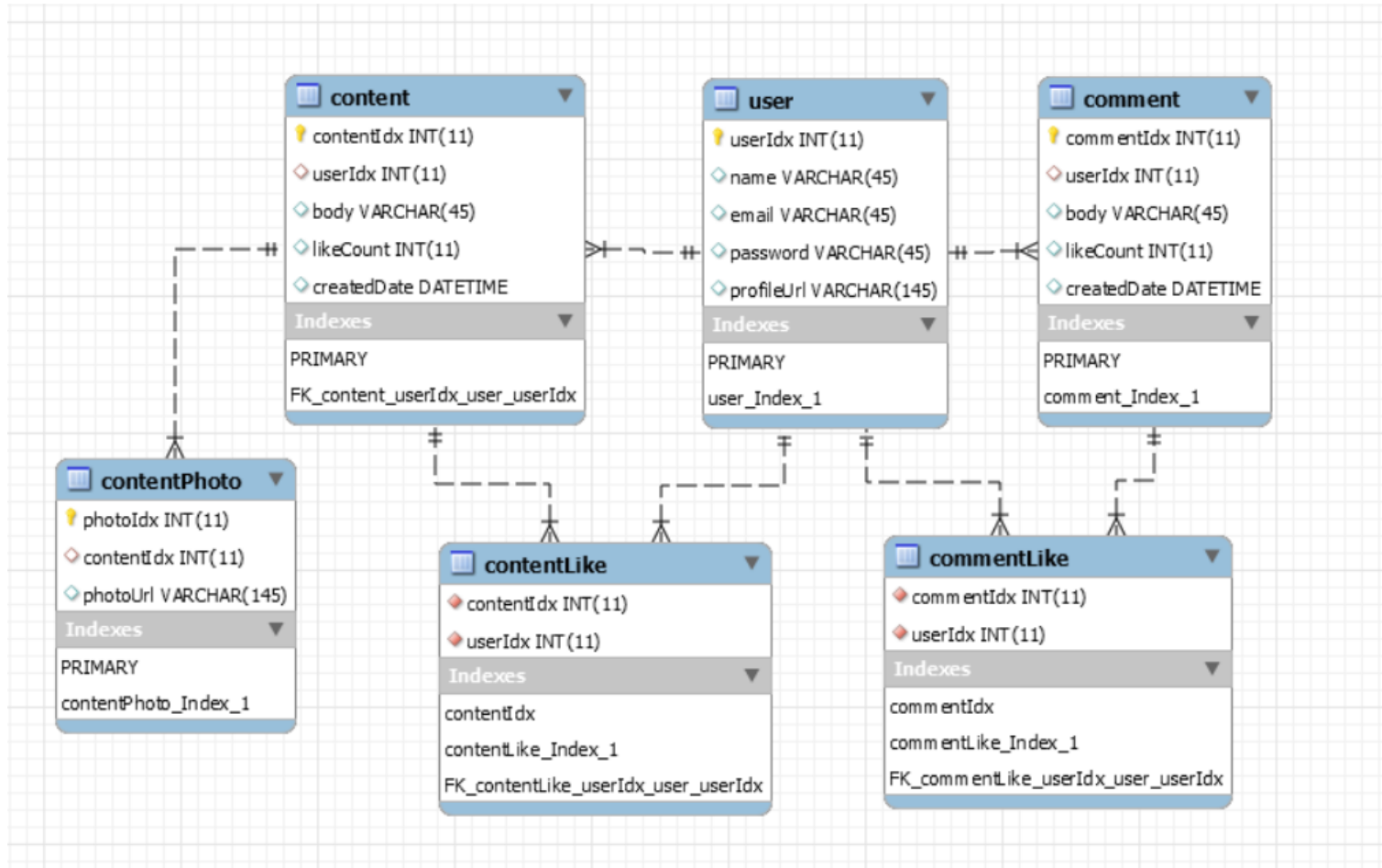
클라이언트와 합동 세미나

1. 통신 연결 성공
2. URI 설계 & API 문서 작성 연습
3. DB 모델링은 제공합니다.
4. 다 완성 못해도 됩니다.
5. 나중에 혼자서 완성을 목표로 개발하는 것을 권장 합니다.

기능

1. 숏스타그램(SOPStagram)
2. 친구기능은 없음 - 모든 콘텐츠를 조회하면 모든 사람이 작성한 콘텐츠가 조회된다.
3. 해쉬 태그 없이 그냥 콘텐츠만 작성
4. 사진은 무조건 1장 이상 넣어서 콘텐츠를 작성해야 한다.
5. 로그인
6. 회원 CRUD
7. 콘텐츠 CRUD
8. 콘텐츠 좋아요 & 취소
9. 댓글 CRUD
10. 댓글 좋아요 & 취소

DB



전체 테이블 생성 SQL 링크

조편성

| 1조 | 2조 | 3조 | 4조 | 5조 | 6조 |
|-----|-----|-----|-----|-----|-----|
| 강태경 | 김소연 | 신지우 | 박찬인 | 김현진 | 권현아 |
| 백선헤 | 김현수 | 이종찬 | 임수정 | 송지은 | 김가영 |
| 이정재 | 박소정 | 조민주 | 허수진 | 유현영 | 박다예 |
| 조재유 | 이진우 | 조연주 | | 최선틸 | |
| 7조 | 8조 | 9조 | 10조 | 11조 | 12조 |
| 류수한 | 강수정 | 구설원 | 김현희 | 권서연 | 박경선 |
| 안나연 | 박영우 | 김보미 | 성누리 | 김민정 | 박선영 |
| 유기태 | 설수연 | 문지현 | 이상은 | 백예은 | 원정빈 |
| | 이상윤 | 서영은 | 황유선 | 정명지 | 최수정 |

○

03

○

차후 일정 안내

8차 세미나

1. 2018년 12월 29일
2. Spring Data JPA
3. Interceptor
4. 그동안 했던 거 총 정리
5. 정상적으로 출석 합니다.

앱잼 일정

1. 자율 팀 빌딩 기간 : 12월 8일 토요일 ~ 12월 21일 금요일 18:00
2. 기획안 경선 : 12월 22일 토요일 12:00 ~ 14:00 서울 창업 허브 10층 대강당
3. 데모 데이 : 12월 22일 토요일 14:00 ~ 18:00 서울 창업 허브 10층 대강당
4. 앱잼 시작 : 12월 22일 토요일 ~ 2019년 1월 12일 토요일
5. 최종 발표 : 2019년 1월 12일 토요일 12:00 ~ 18:00 MARU180 지하 1층 이벤트홀

앱잼 이후 일정

1. 2019년 1월 19일 ~ 1월 20일 후반기 MT

2. 2019년 1월 16일 ~ 1월 23일 서버 파트장 입후보
기간

3. 2019년 1월 26일 종무식 및 23대 임원진 선거

S H O U T · O U R · P A S S I O N · T O G E T H E R

ODo IT SOPT O

THANK U

PPT 디자인 한승미 세미나 자료 배다슬

SHOUT OUR PASSION TOGETHER
SOPT