

Spring & Mybatis Project

LElsurvation by 와장창

박찬미

남기문

이승준

이수민

목차

01

프로젝트
기획

02

시연

03

핵심기술
설명

04

소감

INVEON

1. 프로젝트 기획

- 주제 선정

Leisure + Reservation

=

LEIsurvation

1. Origin of the project

- 기획

레저예약

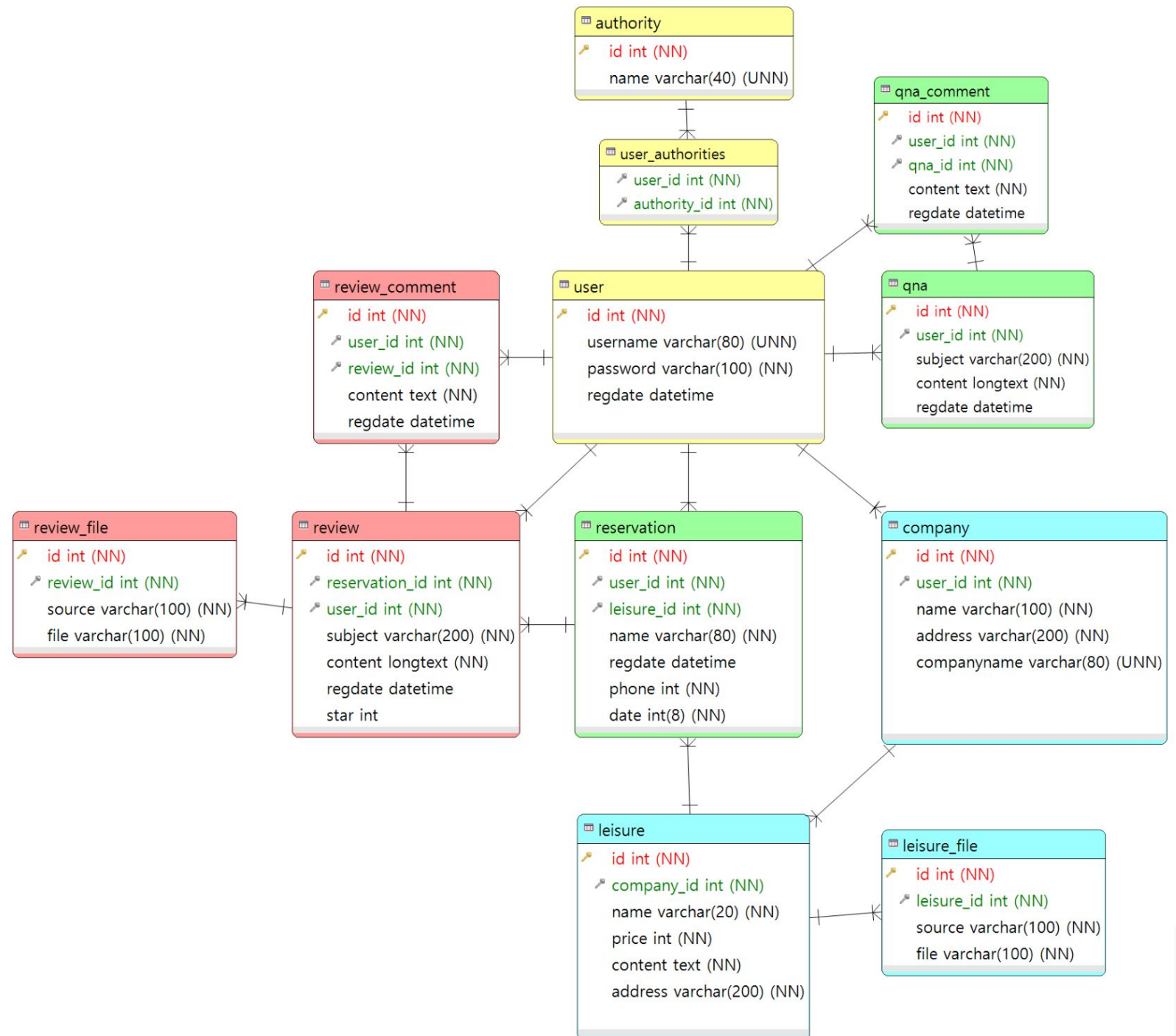
후기작성

문의글
작성

User가
Company
권한 얻음

레저
등록

01. ERD



02. 역할

박찬미

- 권한 인증
- 권한 인가
- 헤더/홈

남기문

- 레저
- 회사
- 레저 파일

이승준

- qna
- qna 댓글
- 레저 예약

이수민

- 리뷰
- 리뷰 댓글
- 리뷰 파일

03. 기술 스택



1. 프로젝트 기획

2. 시연

3. 핵심 기술 설명

- 권한인증
- 권한인가
- 헤더/훅

권한인증(로그인)

- 카카오API

- 1) js를 통해 apiLogin 페이지로 넘어
- 2) hidden 타입인 아이디, 비밀번호
- 3) 기존 카카오 id값이 아닌 경우, apiJoin페이지로 넘어감
- 4) provider가 api인 경우엔 로그인까지 실행

```
if (user.getProvider().equals("api")) {
    model.addAttribute( attributeName: "username", user.getUsername());
    System.out.println("-----");
    System.out.println(user.getUsername());
    System.out.println("-----");
    return "/user/apiLogin";
}

model.addAttribute( attributeName: "result", cnt);
return "/user/joinOk";
}
```

Usercontroller-joinOk()

```
@PostMapping("/apiLogin")
public String apiLogin(String id, Model model){
    model.addAttribute( attributeName: "username", id);

    if(userService.isExist(id)){
        return "/user/apiLogin";
    }else{
        return "/user/apiJoin";
    }
}
```

Usercontroller-apiLogin()

join.html

```
<input type="hidden" name="provider" th:value="general">
```

apiLogin.html

```
<form id="frm" th:action="@{/user/login}" method="POST">
    <input type="hidden" name="username" th:value="${username}">
    <input type="hidden" name="password" th:value="${username.toUpperCase()}">
</form>
```

apiJoin.html

```
<form id="frm" th:action="@{/user/join}" method="POST">
    <input type="hidden" name="username" th:value="${username}">
    <input type="hidden" name="password" th:value="${username.toUpperCase()}">
    <input type="hidden" name="provider" th:value="api">
```

JS

```
Kakao.init( ); //발급받은 키 중 javascript키를 사용하
//console.log(Kakao.isInitialized()); // sdk초기화여부판단
kakaoLogout();

//카카오로그인
function loginWithKakao(){
    Kakao.Auth.login({
        success: function(response){
            Kakao.API.request({
                url: '/v2/user/me',
                success: function(response){
                    let kakaoForm = document.createElement("form");
                    kakaoForm.setAttribute("charset", "UTF-8");
                    kakaoForm.setAttribute("method", "post");
                    kakaoForm.setAttribute("action", "apiLogin");
                    document.body.appendChild(kakaoForm);

                    console.log(response);

                    const userId = "kakao" + response.id;
                    console.log(userId);

                    const input1 = document.createElement('input');
                    input1.type = "hidden"; input1.name = "id"; input1.value = userId;
                    kakaoForm.append(input1);

                    kakaoForm.submit();
                }
            });
        }
    });
}
```

권한인증(회원가입)

- **User**필드 **username** : 공백일 경우, error reject
- **User**의 필드 **password** : 정규표현식과 맞지 않을 경우, error reject

DTO

```
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class User {
    private Long id;
    private String username;

    @JsonIgnore
    private String password;
    @JsonIgnore
    private LocalDateTime regdate;
```

Validator

```
User user = (User)target;

String username = user.getUsername();
if(username == null || username.trim().isEmpty()) {
    errors.rejectValue( field: "username", errorCode: "※ username 은 필수입니다");
}

ValidationUtils.rejectIfEmptyOrWhitespace(errors, field: "password", errorCode: "password 는 필수입니다");

String password = user.getPassword();
String p_regex = "(?=.*[a-zA-z])(?=.*[0-9]).{8,}$";
if(!Pattern.matches(p_regex, password)){
    errors.rejectValue( field: "password", errorCode: "※ 비밀번호를 최소 8자리 이상, 영문/숫자 조합으로 입력해주세요.");
}
```

권한인가

1. company/write.html 에서 등록
2. companyJoin 메소드 통해 권한 부여
3. 컨트롤러에 result에 추가
4. 재로그인 통해 완전한 권한 얻음

CompanyController-writeOk()

```
}  
model.addAttribute( attributeName: "result", companyService.companyWrite(companyWrite));  
model.addAttribute( attributeName: "result", companyService.companyJoin(companyWrite));  
model.addAttribute( attributeName: "dto", companyWrite);  
return "company/writeOK";
```

CompanyService

```
public int companyJoin(CompanyWrite companyWrite){  
    // 컴퍼니 권한부여  
    Authority auth = authorityRepository.findByName("ROLE_COMPANY");  
  
    String user_id = companyWrite.getUser_id();  
    Long auth_id = auth.getId();  
    authorityRepository.addAuthority(Long.valueOf(user_id), auth_id);  
  
    return 1;  
}
```

HTML

```
<script th:if="${result > 0}" th:inline="javascript">  
    alert("등록 성공. 재로그인하세요!");  
    location.href = "/user/logout";  
</script>  
</html>
```


헤더/홈

- **헤더** : 익명상태일때 login버튼
로그인된 상태 해당아이디와와
logout버튼
- **홈** : home_image.js에
showImage(), changePic() 통해
이미지 자동변경 동작

Header

```
<a class="btn btn-primary" sec:authorize="isAnonymous()" th:href="@{/user/login}">LOGIN</a>
<form th:action="@{/user/logout}" method="POST" sec:authorize="isAuthenticated()" th:with="user=${#authentication.getPrincipal().getUser()}">
  <!--TODO : 로그아웃후 다시 돌아오기 -->
  <span class="d-flex">
    <span><a class="nav-link" sec:authorize="hasAnyRole('USER', 'COMPANY')" th:href="@{/reserve/member(user_id=${user.id})}">mypage</a></span>
    <span class="text-light p-2"><span>[[${user.username}]]</span>님 환영합니다!</span>
    <span><button class="btn btn-danger" type="submit">Logout</button></span>
  </span>
```

```
<button onclick="changePic(0);" id="btnB"></button>

<button onclick="changePic(1);" id="btnF"></button>
```

JS

```
var num = 1;
function showImage(){
  var num = Math.round(Math.random()*4);
  var objImg = document.getElementById("photo");
  objImg.src = imgArray[num];
  setTimeout(showImage, 1500);
}
function changePic(idx){
  if(idx){
    if(num == 5) return;
    num++;
    var objImg = document.getElementById("photo");
    objImg.setAttribute("src", "img/leisure" + num + ".jpg");
  }else{
    if(num == 1) return;
    num--;
    var objImg = document.getElementById("photo");
    objImg.setAttribute("src", "img/leisure" + num + ".jpg");
  }
}
```

Home

```
<body onload="showImage()">
<!--/* 인증 헤더 */-->
<th:block th:insert="~{common/header::header}"></th:block>
<script src="js/home_image.js"></script>
```

- 레저
- 회사
- 레저 파일

레저 평점

1. review의 평균을 구해야 한다.
→ sql AVG()
2. 테이블 leisure에서 활용할 수 있어야 한다.
→ Left Join
3. 레저 id별로 구분하여야 한다.
→ Group By
4. Null값이 오면 안된다.
→ IfNull

Mapper

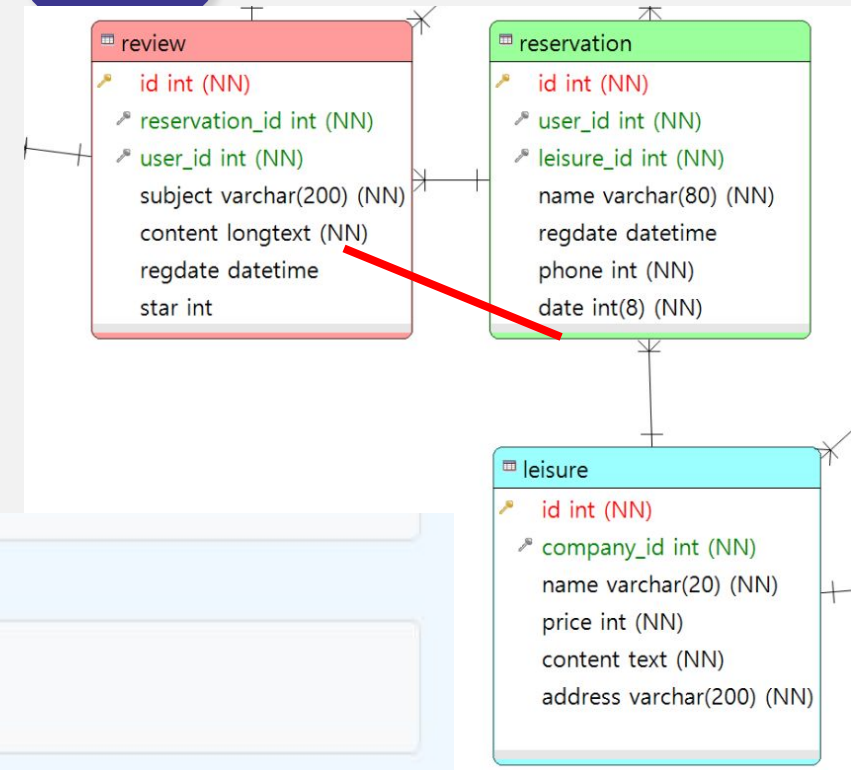
```
<select id="selectAvgStar" resultType="java.lang.Double">
    SELECT IFNULL(AVG(review.star),0) AS avg_star
    FROM leisure      LEFT JOIN reservation ON reservation.leisure_id = leisure.id
    |                LEFT JOIN review ON review.reservation_id = reservation.id
    WHERE leisure.id = #{id}
    GROUP BY leisure.id;
</select>
```

평점:

★1.0

기타

ERD



남기문

1. Companyname 부분에 Unique키를 걸어 두어 같은 이름의 회사를 등록하면 에러 발생
-> Validator를 걸어 Companyname값이 중복된다면 에러 코드 출력 후 리다이렉트

Service

```
private CompanyWrite findByCompanyname(String companyname) {  
    return companyWriteRepository.findBycompanyname(companyname);  
}  
  
public boolean isExist(String companyname){  
    CompanyWrite companyWrite = findByCompanyname(companyname);  
    return (companyWrite != null) ? true : false;  
}
```

Validator

```
if(!result.hasFieldErrors("companyname") && companyService.isExist(companyWrite.getCompanyname())  
    result.rejectValue(field: "companyname", errorCode: "이미 존재하는 회사입니다.");  
}
```

레저 파일

남기문

1. 파일 저장
2. 이미지 파일 Db에 저장
3. 저장된 이미지 파일 불러오기

write

레저 등록

파일 선택

양양이.png

삭제

이미지 등록

detail

g



list



g w4

상
세
정
보

- *qna*
- *qna* 댓글
- *reservation*

고객센터(qna)

qna/list 를 만들던 중
작성자가 보여야 하므로
username 을 user table
에서 가져옴

qna/repository.xml

```
<select id="selectFromRow" resultMap="findName">
    SELECT
        d.id "d_id",
        d.user_id "d_user_id",
        d.subject "d_subject",
        d.content "d_content",
        d.regdate "d_regdate",
        s.id "s_id",
        s.username "s_username"
    FROM
        qna d, user s
    WHERE
        d.user_id =s.id
    ORDER BY d.id DESC
    LIMIT #{param1},#{param2}
</select>
```

qna/list

#	제목	등록자	등록일
16	sad	COMPANY1	2023-03-04 13:06:55

```
<tr th:each="dto:${list}">
    <td>[[${dto.id}]]</td>
    <td>[[${dto.subject}]]</td>
    <td>[[${dto.user.username}]]</td>
    <td>[[${#temporals.format(dto.regdate, 'yyyy-MM-dd HH:mm:ss')}]]</td>
    <td><a class="btn btn-outline-dark" th:href="@{/qna/detail(id=${dto.id})}">상세보기</a></td>
</tr>
```

qna/list.html

고객센터(qna)

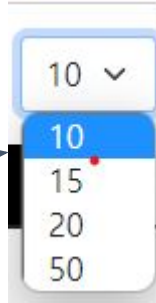
1. 현재 페이지를 page로 받고, 세션에 저장한다.

2. 기본 페이징 세팅값은 세션에 저장하고 기본값은 10으로한다.

qna/controller

```
@GetMapping("/list")
public void list(Integer page, Model model){
    qnaService.list(page, model);
}
```

```
@PostMapping("/pageRows")
public String pageRows(Integer page, Integer pageRows){
    U.getSession().setAttribute(s: "pageRows", pageRows);
    return "redirect:/qna/list?page=" + page;
}
```



qna/service

```
public List<Qna> list(Integer page, Model model){
    //현재 페이지 parameter
    if(page==null) page=1;
    if(page < 1) page=1;

    HttpSession session = U.getSession();
    Integer writePages=(Integer)session.getAttribute(s: "writePages");
    if(writePages == null) writePages = C.WRITE_PAGES;
    Integer pageRows=(Integer)session.getAttribute(s: "pageRows");
    if(pageRows == null) pageRows = C.PAGE_ROWS;

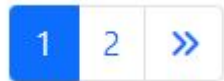
    session.setAttribute(s: "page", page); //세션에 현재 페이지가 몇 페이지인지 저장한다.

    long cnt = qnaRepository.countAll(); // 글 목록 전체의 개수
    int totalPage = (int)Math.ceil(cnt/(double)pageRows); //총 몇 페이지 분량인가

    // page 값 보정
    if(page > totalPage) page = totalPage;

    // 몇 번째 데이터부터 fromRow
    int fromRow = (page - 1) * pageRows;

    // [페이징] 에 표시할 '시작페이지' 와 '마지막페이지' 계산
    int startPage = ((int)((page - 1) / writePages) * writePages) + 1;
    int endPage = startPage + writePages - 1;
    if (endPage >= totalPage) endPage = totalPage;
}
```



기본 세팅 값

```
public class C {

    // 페이징 관련 세팅값
    public static final Integer WRITE_PAGES = 10;
    public static final Integer PAGE_ROWS = 10;
}
```

고객센터(qna/detail)댓글목록

댓글 입력 정보는 JSON 으로
받아와져서 사용됨, ajax 사용,
삭제버튼 활성화

json

```
{
  "count": 6,
  "status": "Ok",
  "data": [{
    "id": 18,
    "content": "18. admin1이 1번글에 댓글 작성.",
    "user": {
      "id": 4,
      "username": "ADMIN1"
    },
    "regdate": "2023-03-01 20:51:16"
  }, {
```

qnaComment/deta
il.js

```
const delBtn = (logged_id !== user_id) ? '' : `
  <i class="btn fa-solid fa-delete-left text-danger" data-bs-toggle="tooltip"
    data-cmtid=${id}" title="삭제"></i>
  `;

```


```
<select id="findByWrite" resultMap="mapComment">
  SELECT
    c.id "c_id",
    c.content "c_content",
    c.regdate "c_regdate",
    c.qna_id "c_qna_id",
    u.id "u_id",
    u.username "u_username",
    u.password "u_password",
    u.regdate "u_regdate"
  FROM qna_comment c, user u
  WHERE
    c.user_id = u.id
    AND c.qna_id = #{param1}
  ORDER BY c.id DESC
</select>
```

COMPANY1

9. company1이 1번글에 댓글 작성.

2023-03-04 17:19:34

USER1

2. user1이 1번글에 댓글 작성. 

2023-03-04 17:19:34

reservation 의 mypage

1. 리스트를 2개 받아와서 각각 예약목록과 해당 유저가 갖고 있는 회사 정보를 알려준다.
2. 프로필의 정보를 예약목록이나 회사목록 테이블에 의존 -> 둘 중 하나가 없으면 프로필이 나오지 않는 문제가 발생 -> 현재 로그인한 username 을 가져오는 방식으로 해결
3. thymeleaf의 if 문을 써서 목록이 없으면 보이지 않게 설정하여 오류를 없앴다.
4. 한 유저가 여러 개의 회사를 가질 수 있다는 사실 간과 -> Join을 사용하여 reserve/company 로 들어가는 url 을 본래의 user_id 가 아닌 company_id 로 받아오는 것으로 바꿈.

reserve/controller

```
@GetMapping("/member")
public void list(Long user_id, Model model){
    model.addAttribute("list",reserveService.list(user_id));
    model.addAttribute("list2",reserveService.won(user_id));
}
```

reserve/member.html

```
<th:block th:if="${list2} and ${#lists.size(list2)} > 0">
```

```
<div style="background-color:cornflowerblue" id="memhead">
  <h3>프로필</h3>
  <div class="mb-3" th:with="user=${#authentication.getPrincipal().getUser()}">유저 아이디: [[${user.username}]]</div>
</div>
```

reserve/repository.xml

```
<select id="findAll2" resultMap="mapCom1">
  SELECT
    c.id "c_id",
    c.name "c_name",
    c.companyname "c_companyname",
    c.address "c_address",|
    u.id "user_id",
    u.username "u_username"
  FROM
    company c, user u
  WHERE u.id= c.user_id
    AND u.id = #{user_id}
  ;
</select>
```


- 리뷰
- 리뷰 댓글
- 리뷰 파일

Validator의 검사 대상

1. 권한 인증을 위해 User 객체를 사용
2. Validator가 Review 안의 User 또한 검증 할 객체인지 조사
3. 아래와 같이 User또한 검사 대상으로

DTO

```
public class Review {  
    private Long id;  
    private String reservation_id;  
    private String subject;  
    private String content;  
    private LocalDateTime regdate;  
    private int star;  
  
    // 글 작성자(FK)  
    private User user_id;  
}
```

Validator

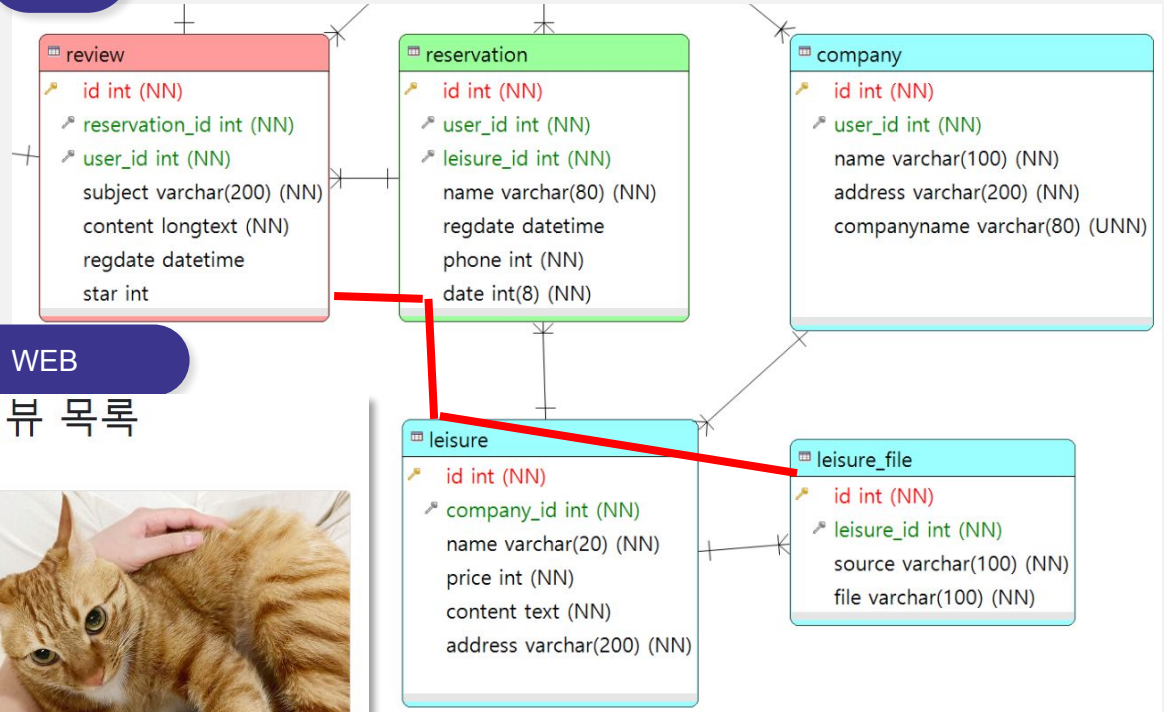
```
@Override  
public boolean supports(Class<?> clazz) {  
    boolean result = Review.class.isAssignableFrom(clazz) || User.class.isAssignableFrom(clazz) ;  
    System.out.println(result);  
    return result;  
}
```

리뷰

이수민

1. review 페이지에서 leisure_file이 필요
2. 레저 id별로 file이 필요 → Group By
3. 교집합인 부분만 → LEFT JOIN으로 null 값이 담긴다 → leisure_file이 없어도 열람 가능

DB



WEB

리뷰 목록



양양리조트

★5

자세히보기 →

xml

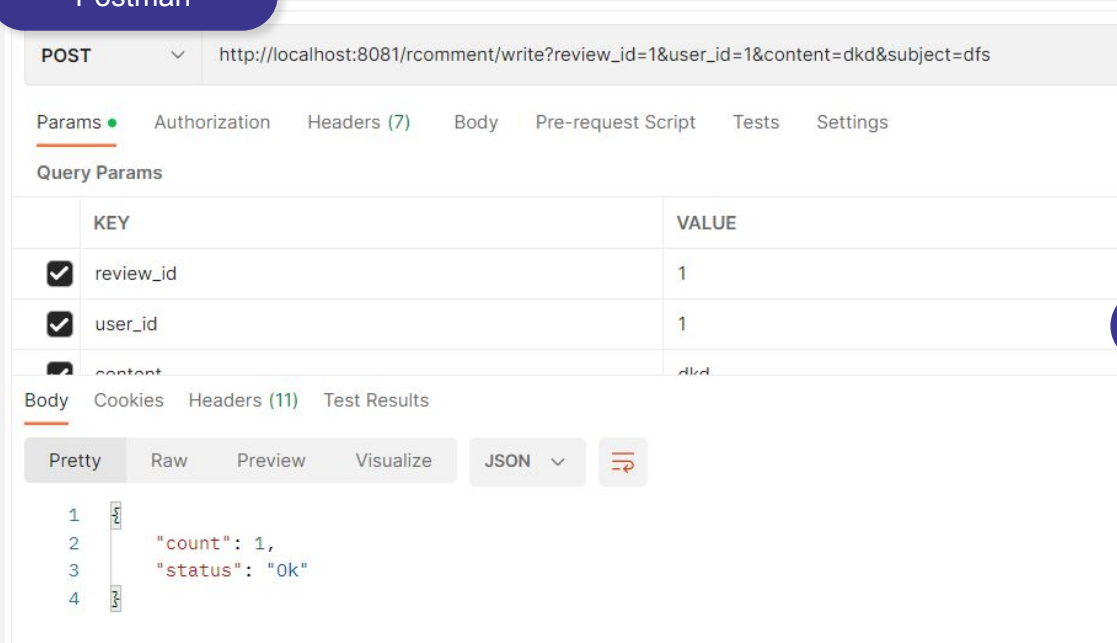
```
<select id="findAll" resultType="com.lec.spring.domain.Review">
    SELECT r.id, reservation_id, r.user_id, subject, r.content, r.regdate, star, l.name AS leisure_name, rs.`date` AS reserve_date, rs.name AS rese
        l.id AS leisure_id, lf.file AS min_leisure_file
    FROM review r INNER JOIN reservation rs ON (r.reservation_id = rs.id)
        INNER JOIN leisure l ON (rs.leisure_id = l.id)
        LEFT JOIN (SELECT leisure_id, min(id) AS leisure_file_id FROM leisure_file GROUP BY leisure_id) m_lf ON (l.id = m_lf.leisure_id)
        LEFT JOIN leisure_file lf ON (lf.leisure_id = m_lf.leisure_id AND lf.id = m_lf.leisure_file_id)
    ORDER BY r.id DESC;
</select>
```

리뷰 댓글

이수민

1. @RequestParam으로 js에서 dto의 역할을 할 수 있게 한다.
2. controller를 만든후 test를 위해 Postman을 해본다 → 반환값을 QryResult DTO를 이용하여 프로토타입없이 확인한다.

Postman



CommentController

```
public RQryResult write(
    @RequestParam("review_id") Long review_id,
    @RequestParam("user_id") Long user_id,
    String content) {
    return commentService.write(review_id,user_id,content);
}
```

detail.js

```
// 전달할 parameter들 준비
const data = {
  "review_id" : id,
  "user_id" : logged_id,
  "content" : content
};
```

QryResult

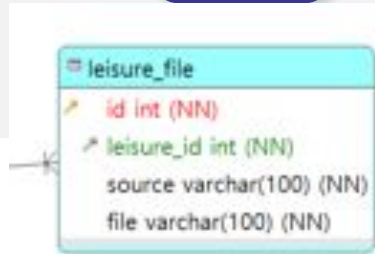
```
public class RQryResult {
    int count; // 결과값 (정수)
    String status; // 결과 메시지
}
```

리뷰 파일

이수민

1. 최상위 주소를 Config에서 설정
2. 두 개의 파일이 필요 → 저장공간의 분리 필요 예) upload/review/[저장된 파일명]
3. review_id 어느글의 file

DB



ConfigFile

```
@Override
public void addResourceHandlers(ResourceHandlerRegistry registry) {
    registry.addResourceHandler(...pathPatterns: "/upload/**")
        .addResourceLocations("file:upload/");
}
```

DTO

```
private String source; //업로드 파일명
private String file; //저장된 파일명
```

```
# 파일 업로드 경로
app.leisure.upload.path=upload/leisure

# 리뷰 파일 경로
app.review.upload.path=upload/review
```

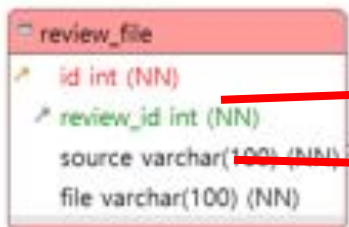
Repository

```
mapper namespace="com.lec.spring.repository.ReviewFileRepository">
    <insert id="insert" flushCache="true" parameterType="arraylist">
        INSERT INTO review_file(source, file, review_id)
        VALUES
        <foreach collection="param1" item="m" separator=",">
            (#{m.source}, #{m.file}, #{param2})
        </foreach>
    </insert>
```

```
int insert(List<Map<String, Object>> list, Long review_id);

int save(ReviewFileDTO file);

// 특정 글(review_id)의 첨부파일들
List<ReviewFileDTO> findByReview(Long review_id);
```



```
int insert(List<Map<String, Object>> list, Long review_id);
```

4. 소감

박찬미

남기문

이승준

이수민

THANK YOU



LIVE ON