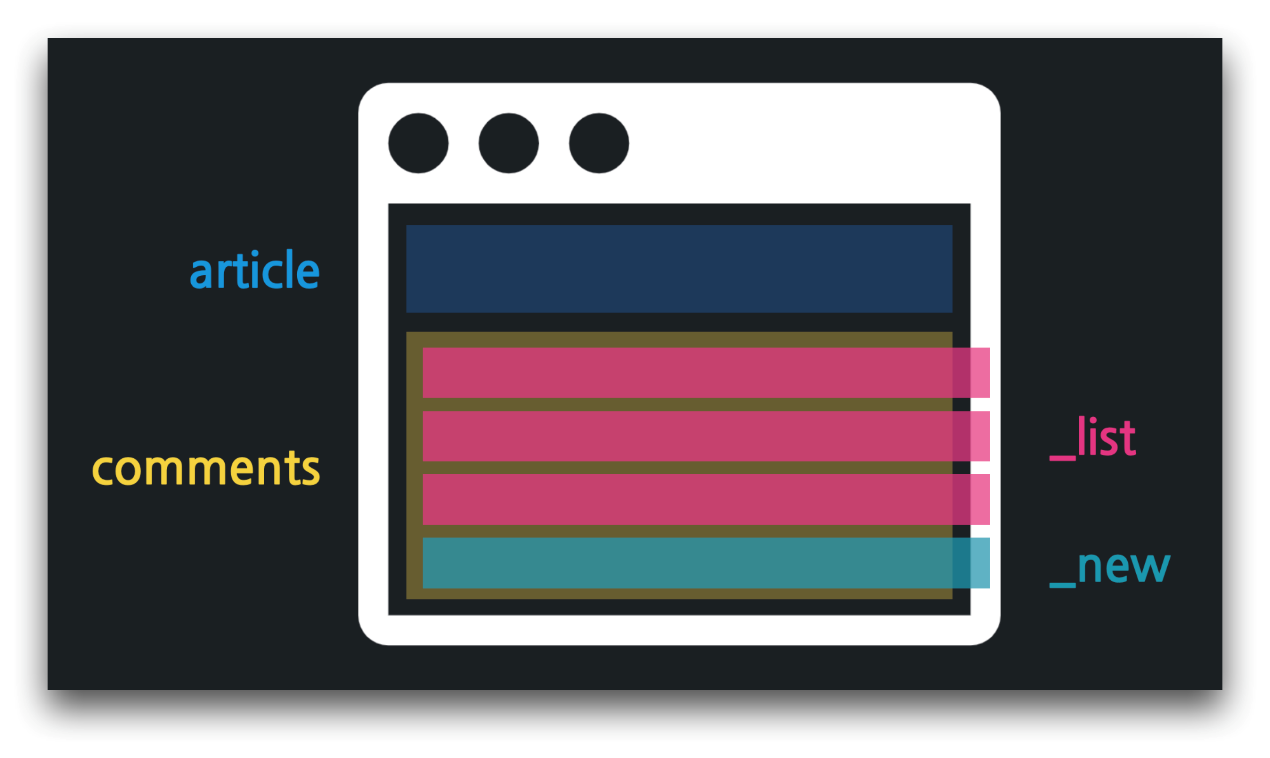
24 댓글 목록 뷰 만들기

**댓글 목록 뷰 만들기**

**미션**

댓글 목록을 게시글 상세 페이지에 추가하시오.  


**01:17 댓글 뷰 페이지 삽입 - mustache**

../articles/show.mustache

1. {{>layouts/header}}
2. <table class="table">
3. ...
4. </table>
5. <a href="/articles/{{article.id}}/edit" class="btn btn-primary">Edit</a>
6. <a href="/articles/{{article.id}}/delete" class="btn btn-danger">Delete</a>
7. <a href="/articles">Go to Article List</a>
8. {{>comments/\_comments}}
9. {{>layouts/footer}}

../comments/\_comments.mustache

1. <div>
2. <!-- 댓글 목록 -->
3. {{>comments/\_list}}
4. <!-- 새 댓글 작성 -->
5. {{>comments/\_new}}
6. </div>

../comments/\_list.mustache

1. <div id="comments-list">
2. {{#commentDtos}}
3. <div class="card m-2" id="comments-{{id}}">
4. <div class="card-header">
5. {{nickname}}
6. </div>
7. <div class="card-body">
8. {{body}}
9. </div>
10. </div>
11. {{/commentDtos}}
12. </div>

../comments/\_new.mustache

1. <!-- 파일만 생성 -->

**07:25 댓글 목록 가져오기 - Controller, Service, Model, View, addattribute**

../controller/ArticleController

1. ...
2. public class ArticleController {
3. @Autowired
4. private ArticleRepository articleRepository;
5. @Autowired
6. private CommentService commentService;
7. ...
8. @GetMapping("/articles/{id}") // 해당 URL요청을 처리 선언
9. public String show(@PathVariable Long id,
10. Model model) { // URL에서 id를 변수로 가져옴
11. log.info("id = " + id);
12. // 1: id로 데이터를 가져옴!
13. Article articleEntity = articleRepository.findById(id).orElse(null);
14. List<CommentDto> commentsDtos = commentService.comments(id);
15. // 2: 가져온 데이터를 모델에 등록!
16. model.addAttribute("article", articleEntity);
17. model.addAttribute("commentDtos", commentsDtos);
18. // 3: 보여줄 페이지를 설정!
19. return "articles/show";
20. }
21. ...
22. }

**🔥 구글링 훈련하기**

* 없음