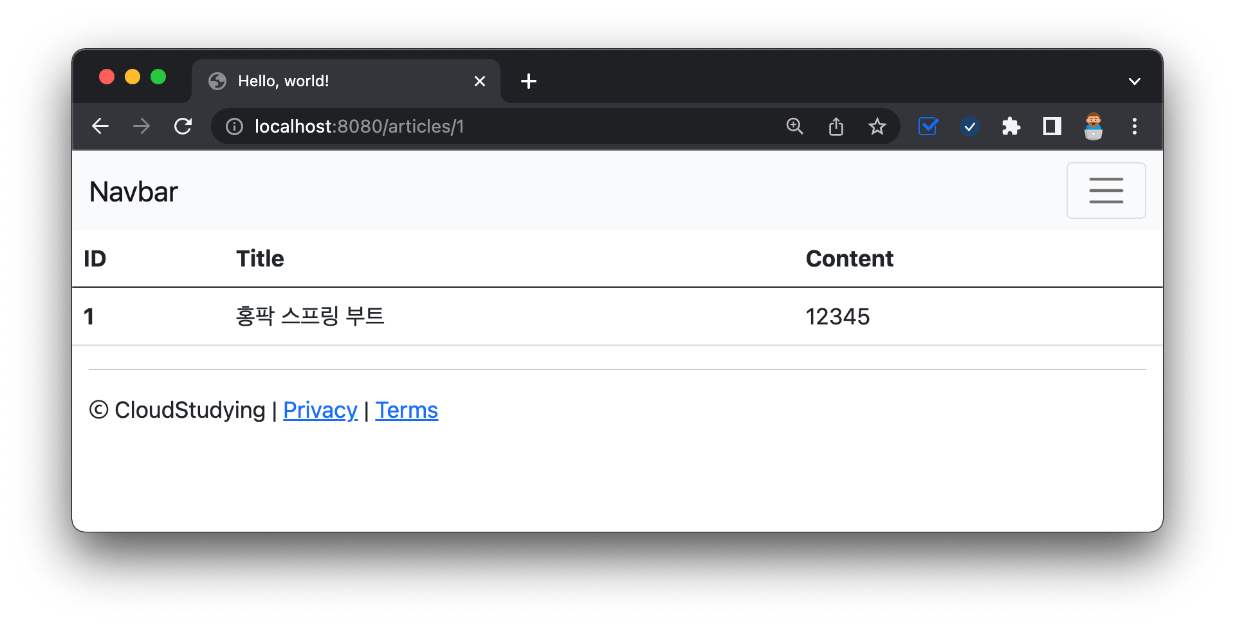
11 데이터 조회(read)하기 with JPA

**데이터 조회하기 with JPA**

**미션**

DB속 데이터를 조회하여, 웹 페이지로 확인하시오.  


**URL 요청받기 - PathVariable**

../controller/ArticleController

1. ...
2. @Controller
3. @Slf4j
4. public class ArticleController {
5. ...
6. @GetMapping("/articles/{id}") // 해당 URL요청을 처리 선언
7. public String show(@PathVariable Long id) { // URL에서 id를 변수로 가져옴
8. log.info("id = " + id);
9. return "";
10. }
11. }

**컨트롤러 - 처리 흐름**

../controller/ArticleController

1. ...
2. @Controller
3. @Slf4j
4. public class ArticleController {
5. ...
6. @GetMapping("/articles/{id}") // 해당 URL요청을 처리 선언
7. public String show(@PathVariable Long id) { // URL에서 id를 변수로 가져옴
8. log.info("id = " + id);
9. // 1: id로 데이터를 가져옴!
10. // 2: 가져온 데이터를 모델에 등록!
11. // 3: 보여줄 페이지를 설정!
12. return "";
13. }
14. }

**리파지터리 - id 조회, findById()**

../controller/ArticleController

1. ...
2. public class ArticleController {
3. ...
4. @GetMapping("/articles/{id}") // 해당 URL요청을 처리 선언
5. public String show(@PathVariable Long id,
6. Model model) { // URL에서 id를 변수로 가져옴
7. log.info("id = " + id);
8. // 1: id로 데이터를 가져옴!
9. Article articleEntity = articleRepository.findById(id).orElse(null);
10. // 2: 가져온 데이터를 모델에 등록!
11. model.addAttribute("article", articleEntity);
12. // 3: 보여줄 페이지를 설정!
13. return "articles/show";
14. }
15. }

**뷰 - mustache 템플릿 작성**

../articles/show.mustache

1. {{>layouts/header}}
2. <table class="table">
3. <thead>
4. <tr>
5. <th scope="col">ID</th>
6. <th scope="col">Title</th>
7. <th scope="col">Content</th>
8. </tr>
9. </thead>
10. <tbody>
11. {{#article}}
12. <tr>
13. <th>{{id}}</th>
14. <td>{{title}}</td>
15. <td>{{content}}</td>
16. </tr>
17. {{/article}}
18. </tbody>
19. </table>
20. {{>layouts/footer}}

**디버깅 - 디폴트 생성자 추가, NoArgsConstructor**

../entity/Article

1. package com.example.firstproject.entity;
2. import lombok.AllArgsConstructor;
3. import lombok.NoArgsConstructor;
4. import lombok.ToString;
5. import javax.persistence.Column;
6. import javax.persistence.Entity;
7. import javax.persistence.GeneratedValue;
8. import javax.persistence.Id;
9. @Entity
10. @AllArgsConstructor
11. @NoArgsConstructor // 디폴트 생성자 추가
12. @ToString
13. public class Article {
14. @Id
15. @GeneratedValue
16. private Long id;
17. @Column
18. private String title;
19. @Column
20. private String content;
21. }

**🔥 구글링 훈련하기**

* @PathVariable
* JPA CrudRepository findById 사용법
* @NoArgsConstructor
* Java Optional 사용법