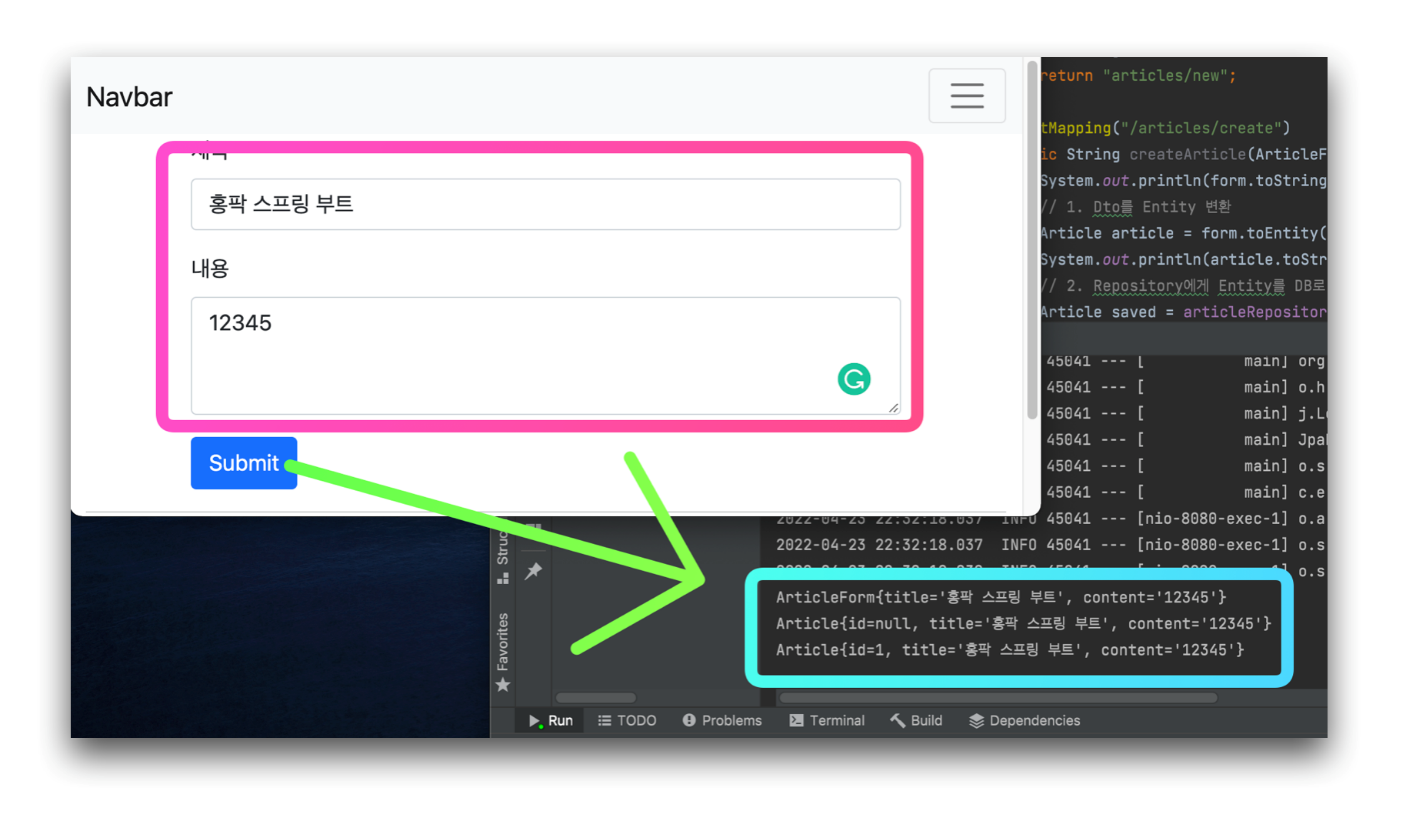
08 데이터 생성 with JPA

**데이터 생성 with JPA**

**미션**

JPA를 활용하여 폼 데이터를 DB로 저장하시오.  


**DTO를 엔티티로 변환하기**

../controller/ArticleController

1. ...
2. @Controller
3. public class ArticleController {
4. ...
5. @PostMapping("/articles/create")
6. public String createArticle(ArticleForm form) {
7. System.out.println(form.toString());
8. // 1. Dto를 Entity 변환
9. Article article = form.toEntity();
10. // 2. Repository에게 Entity를 DB로 저장하게 함
11. return "";
12. }
13. }

**엔티티 작성하기**

../entity/Article

1. package com.example.firstproject.entity;
2. import javax.persistence.Column;
3. import javax.persistence.Entity;
4. import javax.persistence.GeneratedValue;
5. import javax.persistence.Id;
6. @Entity
7. public class Article {
8. @Id
9. @GeneratedValue
10. private Long id;
11. @Column
12. private String title;
13. @Column
14. private String content;
15. public Article(Long id, String title, String content) {
16. this.id = id;
17. this.title = title;
18. this.content = content;
19. }
20. @Override
21. public String toString() {
22. return "Article{" +
23. "id=" + id +
24. ", title='" + title + '\'' +
25. ", content='" + content + '\'' +
26. '}';
27. }
28. }

**변환 메소드 추가**

../dto/ArticleForm

1. package com.example.firstproject.dto;
2. import com.example.firstproject.entity.Article;
3. public class ArticleForm {
4. ...
5. public Article toEntity() {
6. return new Article(null, title, content);
7. }
8. }

**리파지터리를 통한 엔티티 저장하기**

../controller/ArticleController

1. package com.example.firstproject.controller;
2. import com.example.firstproject.dto.ArticleForm;
3. import com.example.firstproject.entity.Article;
4. import org.springframework.stereotype.Controller;
5. import org.springframework.web.bind.annotation.GetMapping;
6. import org.springframework.web.bind.annotation.PostMapping;
7. @Controller
8. public class ArticleController {
9. private ArticleRepository articleRepository;
10. ...
11. @PostMapping("/articles/create")
12. public String createArticle(ArticleForm form) {
13. System.out.println(form.toString());
14. // 1. Dto를 Entity 변환
15. Article article = form.toEntity();
16. // 2. Repository에게 Entity를 DB로 저장하게 함
17. Article saved = articleRepository.save(article);
18. return "";
19. }
20. }

**리파지터리 작성**

../repository/ArticleRepository

1. package com.example.firstproject.repository;
2. import com.example.firstproject.entity.Article;
3. import org.springframework.data.repository.CrudRepository;
4. public interface ArticleRepository extends CrudRepository<Article, Long> {
5. }

**의존성 주입하기(DI)**

../controller/ArticleController

1. package com.example.firstproject.controller;
2. import com.example.firstproject.dto.ArticleForm;
3. import com.example.firstproject.entity.Article;
4. import com.example.firstproject.repository.ArticleRepository;
5. import org.springframework.beans.factory.annotation.Autowired;
6. import org.springframework.stereotype.Controller;
7. import org.springframework.web.bind.annotation.GetMapping;
8. import org.springframework.web.bind.annotation.PostMapping;
9. @Controller
10. public class ArticleController {
11. @Autowired // 스프링 부트가 미리 생성해놓은 리파지터리 객체를 가져옴(DI)
12. private ArticleRepository articleRepository;
13. ...
14. @PostMapping("/articles/create")
15. public String createArticle(ArticleForm form) {
16. System.out.println(form.toString());
17. // 1. Dto를 Entity 변환
18. Article article = form.toEntity();
19. // 2. Repository에게 Entity를 DB로 저장하게 함
20. Article saved = articleRepository.save(article);
21. return "";
22. }
23. }

**데이터 저장 흐름 확인**

../controller/ArticleController

1. package com.example.firstproject.controller;
2. import com.example.firstproject.dto.ArticleForm;
3. import com.example.firstproject.entity.Article;
4. import com.example.firstproject.repository.ArticleRepository;
5. import org.springframework.beans.factory.annotation.Autowired;
6. import org.springframework.stereotype.Controller;
7. import org.springframework.web.bind.annotation.GetMapping;
8. import org.springframework.web.bind.annotation.PostMapping;
9. @Controller
10. public class ArticleController {
11. @Autowired
12. private ArticleRepository articleRepository;
13. @GetMapping("/articles/new")
14. public String newArticleForm() {
15. return "articles/new";
16. }
17. @PostMapping("/articles/create")
18. public String createArticle(ArticleForm form) {
19. System.out.println(form.toString());
20. // 1. Dto를 Entity 변환
21. Article article = form.toEntity();
22. System.out.println(article.toString());
23. // 2. Repository에게 Entity를 DB로 저장하게 함
24. Article saved = articleRepository.save(article);
25. System.out.println(saved.toString());
26. return "";
27. }
28. }

**🔥 구글링 훈련하기**

* JPA 개념
* JPA 리파지터리란
* JPA Entity 사용법
* @Entity
* @Table
* @Id
* @GeneratedValue
* @Column
* DTO Entity 분리 장점
* JPA CrudRepository
* @Autowired