3 장 완성 코드

templates/new.mustache

```
{{>layouts/header}}

<form class="container" action="/articles/create" method="post">
    <div class="mb-3">
        <label class="form-label">제목</label>
        <input type="text" class="form-control" name="title">
    </div>
    <div class="mb-3">
        <label class="form-label">내용</label>
        <textarea class="form-control" rows="3" name="content"></textarea>
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
</form>

{{>layouts/footer}}
```

controller/ArticleController.java

```java
package com.example.firstproject.controller;

import com.example.firstproject.dto.ArticleForm;
import com.example.firstproject.entity.Article;
import com.example.firstproject.repository.ArticleRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;

@Controller
public class ArticleController {
    @Autowired
    private ArticleRepository articleRepository;

    @GetMapping("/articles/new")
    public String newArticleForm() {
        return "articles/new";
    }

    @PostMapping("/articles/create")
    public String createArticle(ArticleForm form) {
        System.out.println(form.toString());

        // 1. DTO를 엔티티로 변환
        Article article = form.toEntity();
        System.out.println(article.toString());

        // 2. 리파지터리로 엔티티를 DB에 저장
        Article saved = articleRepository.save(article);
        System.out.println(saved.toString());

        return "";
    }
}
```

dto/ArticleForm.java

```java
package com.example.firstproject.dto;

import com.example.firstproject.entity.Article;

public class ArticleForm {
    private String title; // 제목을 받을 필드
    private String content; // 내용을 받을 필드

    public ArticleForm(String title, String content) {
        this.title = title;
        this.content = content;
    }

    @Override
    public String toString() {
        return "ArticleForm{" +
                "title='" + title + '\'' +
                ", content='" + content + '\'' +
                '}';
    }

    public Article toEntity() {
        return new Article(null, title, content);
    }
}
```

entity/Article.java

```java
package com.example.firstproject.entity;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.Id;

@Entity
public class Article {
    @Id
    @GeneratedValue
    private Long id;

    @Column
    private String title;

    @Column
    private String content;

    // Article 생성자 추가
    public Article(Long id, String title, String content) {
        this.id = id;
        this.title = title;
        this.content = content;
    }

    // toString() 메서드 추가
    @Override
    public String toString() {
        return "Article{" +
                "id=" + id +
                ", title='" + title + '\'' +
                ", content='" + content + '\'' +
                '}';
    }
}
```

repository/ArticleRepository.java

```java
package com.example.firstproject.repository;

import com.example.firstproject.entity.Article;
import org.springframework.data.repository.CrudRepository;

public interface ArticleRepository extends CrudRepository<Article, Long> {
}
```

resources/application.properties

```properties
server.servlet.encoding.force=true
spring.h2.console.enabled=true
```

셀프체크 정답

controller/MemberController.java

```
package com.example.firstproject.controller;

import com.example.firstproject.entity.Member;
import com.example.firstproject.dto.MemberForm;
import com.example.firstproject.repository.MemberRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;

@Controller
public class MemberController {

    @Autowired
    MemberRepository memberRepository;

    @GetMapping("/signup")
    public String signUpPage() {
        return "members/new";
    }

    @PostMapping("/join")
    public String join(MemberForm memberForm) {
        System.out.println(memberForm.toString());
        Member member = memberForm.toEntity();
        System.out.println(member.toString());
        Member saved = memberRepository.save(member);
        System.out.println(saved.toString());
        return "";
    }
}
```

dto/MemberForm.java

```
package com.example.firstproject.dto;

import com.example.firstproject.entity.Member;

public class MemberForm {
    private String email;
    private String password;

    public MemberForm(String email, String password) {
```

```java
        this.email = email;
        this.password = password;
    }

    @Override
    public String toString() {
        return "User{" +
                "email='" + email + '\'' +
                ", password='" + password + '\'' +
                '}';
    }

    public Member toEntity() {
        return new Member(null, email, password);
    }
}
```

entity/Member.java

```java
package com.example.firstproject.entity;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.Id;

@Entity
public class Member {
    @Id
    @GeneratedValue
    Long id;

    @Column
    String email;

    @Column
    String password;

    public Member(Long id, String email, String password) {
        this.id = id;
        this.email = email;
        this.password = password;
    }

    @Override
    public String toString() {
        return "Member{" +
                "id=" + id +
                ", email='" + email + '\'' +
                ", password='" + password + '\'' +
                '}';
    }
}
```

repository/MemberRepository.java

```
package com.example.firstproject.repository;

import com.example.firstproject.entity.Member;
import org.springframework.data.repository.CrudRepository;

public interface MemberRepository extends CrudRepository<Member, Long> {
}
```

templates/members/new.mustache

```
{{>layouts/header}}

<form class="container" action="/join" method="post">
    <div class="mb-3">
        <label class="form-label">이메일</label>
        <input type="email" class="form-control" name="email">
    </div>
    <div class="mb-3">
        <label class="form-label">비밀번호</label>
        <input type="password" class="form-control" name="password">
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
    <a href="/members">Back</a>
</form>

{{>layouts/footer}}
```