

6 장 완성 코드

templates/articles/index.mustache

```
{{>layouts/header}}

<table class="table">
  <thead>
    <tr>
      <th scope="col">Id</th>
      <th scope="col">Title</th>
      <th scope="col">Content</th>
    </tr>
  </thead>
  <tbody>
    {{#articleList}}
    <tr>
      <th>{{id}}</th>
      <td><a href="/articles/{{id}}">{{title}}</a></td>
      <td>{{content}}</td>
    </tr>
    {{/articleList}}
  </tbody>
</table>

<a href="/articles/new">New Article</a>

{{>layouts/footer}}
```

templates/articles/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>
```

```
<a href="/articles">Back</a>
</form>
```

```
{{>layouts/footer}}
```

controller/ArticleController.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 lombok.extern.slf4j.Slf4j;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;

import java.util.List;

@Slf4j
@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) {
        log.info(form.toString());
        // System.out.println(form.toString());

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

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

        return "redirect:/articles/" + saved.getId();
    }
}
```

```

    }

    @GetMapping("/articles/{id}") // 데이터 조회 요청 접수
    public String show(@PathVariable Long id, Model model) { // 매개변수로 id
        받아오기
        log.info("id = " + id); // id를 잘 받았는지 확인하는 로그 찍기

        // 1. id를 조회하여 데이터 가져오기
        Article articleEntity = articleRepository.findById(id).orElse(null);

        // 2. 모델에 데이터 등록하기
        model.addAttribute("article", articleEntity);

        // 3. 뷰 페이지 반환하기
        return "articles/show";
    }

    @GetMapping("/articles")
    public String index(Model model) {
        // 1. 모든 데이터 가져오기
        List<Article> articleEntityList = articleRepository.findAll();

        // 2. 모델에 데이터 등록하기
        model.addAttribute("articleList", articleEntityList);

        // 3. 뷰 페이지 설정하기
        return "articles/index";
    }
}

```

entity/Article

```

package com.example.firstproject.entity;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.Id;
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.ToString;

@Entity
@ToString
@NoArgsConstructor
@AllArgsConstructor
public class Article {

```

```
@Id
@GeneratedValue
private Long id;

@Column
private String title;

@Column
private String content;

}
```

templates/articles/show.mustache

```
{{>layouts/header}}

<table class="table">
  <thead>
    <tr>
      <th scope="col">Id</th>
      <th scope="col">Title</th>
      <th scope="col">Content</th>
    </tr>
  </thead>
  <tbody>
    {{#article}}
    <tr>
      <th>{{id}}</th>
      <td>{{title}}</td>
      <td>{{content}}</td>
    </tr>
    {{/article}}
  </tbody>
</table>

<a href="/articles">Go to Article List</a>

{{>layouts/footer}}
```

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 lombok.extern.slf4j.Slf4j;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;

@Slf4j
@Controller
public class MemberController {

    @Autowired
    MemberRepository memberRepository;

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

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

    @PostMapping("/join")
    public String join(MemberForm memberForm) {
        log.info(memberForm.toString());
        Member member = memberForm.toEntity();
        log.info(member.toString());
        Member saved = memberRepository.save(member);
        log.info(saved.toString());
        return "redirect:/members/" + saved.getId();
    }

    @GetMapping("/members/{id}")
    public String show(@PathVariable Long id, Model model) {
        Member member = memberRepository.findById(id).orElse(null);
    }
}
```

```

        model.addAttribute("member", member);
        return "members/show";
    }

    @GetMapping("/members")
    public String index(Model model) {
        Iterable<Member> members = memberRepository.findAll();
        model.addAttribute("members", members);
        return "members/index";
    }
}

```

entity/Member.java

```

package com.example.firstproject.entity;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.Id;
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.ToString;

@Entity
@AllArgsConstructor
@NoArgsConstructor
@ToString
@Getter
public class Member {
    @Id
    @GeneratedValue
    Long id;

    @Column
    String email;

    @Column
    String password;
}

```

templates/members/show.mustache

```

{{>layouts/header}}
<table class="table">
  <thead>
  <tr>

```

```

        <th scope="col">Id</th>
        <th scope="col">Email</th>
        <th scope="col">Password</th>
    </tr>
</thead>
<tbody>
    {{#member}}
        <tr>
            <th>{{id}}</th>
            <td>{{email}}</td>
            <td>{{password}}</td>
        </tr>
    {{/member}}
</tbody>
</table>

<a href="/members">Go to member list</a>
{{>layouts/footer}}

```

templates/members/index.mustache

```

{{>layouts/header}}

<table class="table">
    <thead>
        <tr>
            <th scope="col">Id</th>
            <th scope="col">Email</th>
            <th scope="col">Password</th>
        </tr>
    </thead>
    <tbody>
        {{#members}}
            <tr>
                <th>{{id}}</th>
                <td><a href="/members/{{id}}">{{email}}</a></td>
                <td>{{password}}</td>
            </tr>
        {{/members}}
    </tbody>
</table>

<a href="/members/new">회원가입</a>

{{>layouts/footer}}

```
