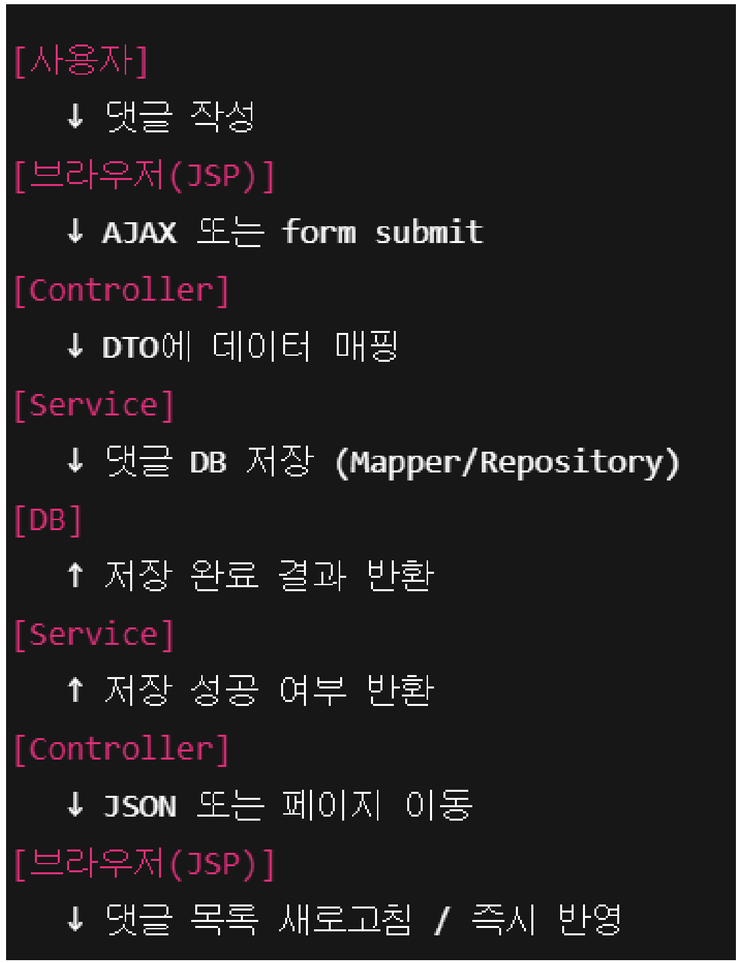
게시판 댓글 구현 프로세스

Spring Boot 교안에 있는 boardProject(게시판)를 기반으로 진행합니다.

프로젝트 자료 깃허브 주소: https://github.com/sjgopromer2025/spring_projects_data

댓글 작성 흐름도


[사용자]
↓ 댓글 작성
[브라우저(JSP)]
↓ AJAX 또는 form submit
[Controller]
↓ DTO에 데이터 매핑
[Service]
↓ 댓글 DB 저장 (Mapper/Repository)
[DB]
↑ 저장 완료 결과 반환
[Service]
↑ 저장 성공 여부 반환
[Controller]
↓ JSON 또는 페이지 이동
[브라우저(JSP)]
↓ 댓글 목록 새로고침 / 즉시 반영

댓글 기능을 추가하려면? 단계별 정리

1. DB 설계 (테이블 생성)
2. DTO 생성
3. Mapper(SQL) 작성
4. Service 작성
5. Controller 작성
6. JSP 뷰 수정
7. JavaScript(AJAX) 작성

comment 테이블 생성

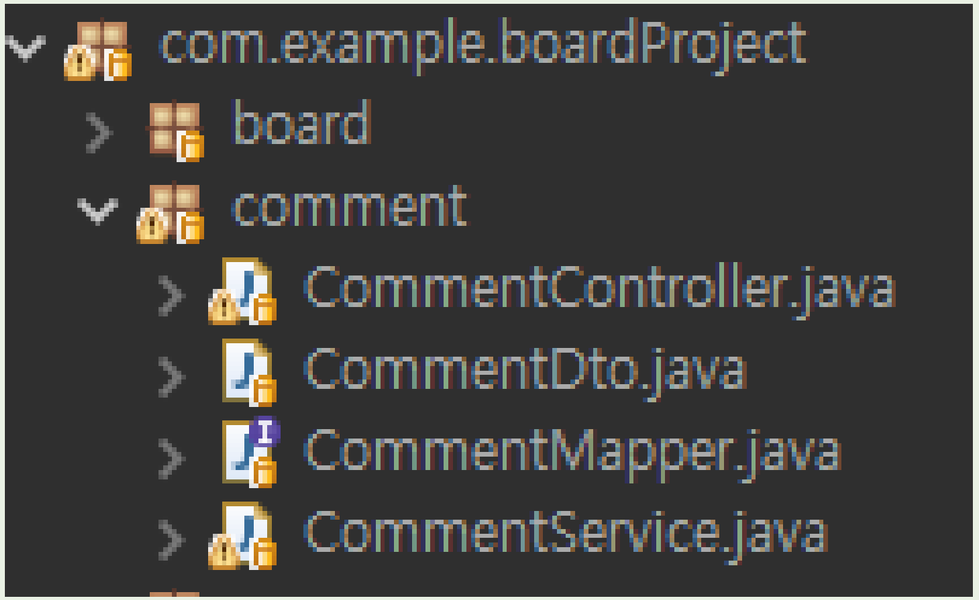
```sql
-- 댓글 정보를 저장하는 'comment' 테이블 생성
CREATE TABLE IF NOT EXISTS practice.comment (
  idx INT AUTO_INCREMENT PRIMARY KEY, -- 댓글 고유 ID
  board_idx INT NOT NULL, -- 해당 댓글이 속한 게시글 ID
  user_name VARCHAR(200) NOT NULL, -- 댓글 작성자 이름
  content TEXT NOT NULL, -- 댓글 내용
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP, -- 작성일
  FOREIGN KEY (board_idx) REFERENCES board(idx) ON DELETE CASCADE -- 외래 키: board 테이블의 idx 참조
);
```

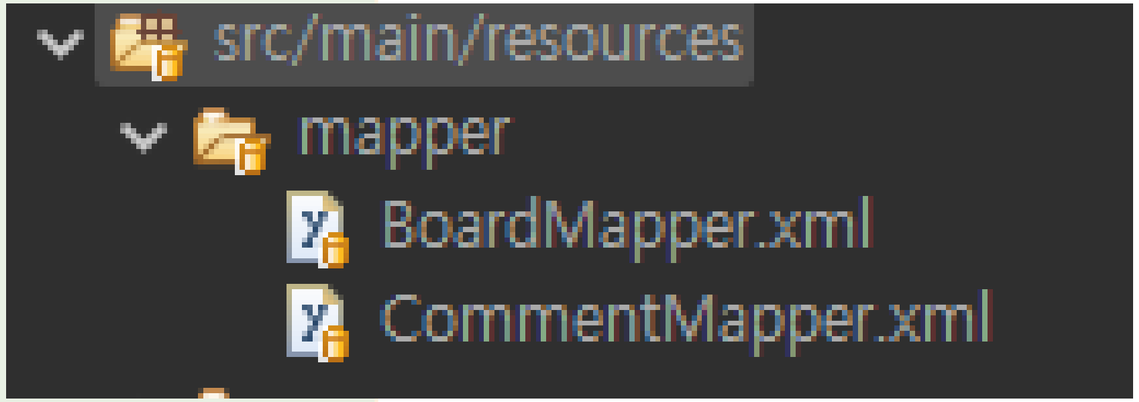
CommnetDTO 생성

```java
public class CommentDto {
    private int idx; // 댓글 고유 ID
    private int boardIdx; // 해당 댓글이 속한 게시글 ID
    private String userName; // 댓글 작성자 이름
    private String content; // 댓글 내용
    private String createdAt; // 작성일

    // getter & setter ~
    // ... 생략
    // toString() ~
    // ... 생략
```

Mapper, Service, Controller 파일 생성





JSP 뷰 수정 - 댓글 작성 영역

```jsp
<section class="comment-container">
  <!-- 댓글 작성 영역-->
  <div class="comment-box">
    <h3>댓글 작성</h3>
    <!-- form 태그 제거 -->
    <input type="hidden" id="boardIdx" value="${board.idx}" />

    <div class="form-group">
      <label for="commentUser">작성자</label>
      <input type="text" id="userName" placeholder="작성자 이름" required />
    </div>

    <div class="form-group">
      <label for="commentContent">내용</label>
      <textarea id="content" rows="4" placeholder="댓글을 입력하세요" required></textarea>
    </div>

    <button type="button" id="submitCommentBtn" class="submit-btn">댓글 등록</button>
  </div>
</section>
```

JSP 뷰 수정 - 댓글 등록 AJAX - 1

```
$("#submitCommentBtn").click(function () {
    const boardIdx = $("#boardIdx").val();
    const userName = $("#userName").val().trim();
    const content = $("#content").val().trim();

    if (!userName || !content) {
      alert("작성자와 내용을 모두 입력해주세요.");
      return;
    }

    const data = {
      boardIdx: boardIdx,
      userName: userName,
      content: content,
    };
```

JSP 뷰 수정 - 댓글 등록 AJAX - 2

```
// 댓글 등록 AJAX 요청
$.ajax({
  url: "/comment/write",     // 요청 URL
  type: "POST",              // 요청 방식
  data: data,                // 폼 데이터 (ex: { userName: "", content: "" })
  success: function (res) {
    // 서버에서 응답 성공
    if (res === "success") {
        location.reload();
    } else {
      // 서버 응답이 실패일 경우
      alert("댓글 등록 실패: " + res);
    }
  },

  error: function (xhr) {
    // AJAX 요청 자체가 실패했을 경우
    alert("에러 발생: " + xhr.responseText);
  }
});
```

JSP 뷰 수정 - 댓글 목록 영역

```
<section class="comment-container">
    <!-- 댓글 목록 영역-->
  <div class="comment-box">
      <h3>댓글 목록</h3>
      <ul class="custom-list">
          <c:choose>
              <c:when test="${empty commentList}">
                  <li id="noneComment">작성된 댓글이 없습니다.</li>
              </c:when>
              <c:otherwise>
                  <c:forEach var="comment" items="${commentList}">
                      <li>
                          <strong>${comment.userName}</strong> (${comment.createdAt})<br/>
                          ${comment.content}
                      </li>
                  </c:forEach>
              </c:otherwise>
          </c:choose>
      </ul>
  </div>
</section>
```

CommentController

```java
@Controller
@RequestMapping("/comment") // "/comment" 경로로 들어오는 요청을 처리
public class CommentController {

    private final CommentService commentService; // 비즈니스 로직을 처리하는 서비스

    @Autowired
    public CommentController(CommentService commentService) {
        this.commentService = commentService;
    }

    @PostMapping("/write")
    @ResponseBody
    public String createComment(CommentDTO commentDto) {
        return commentService.createComment(commentDto) ? "success" : "fail";
    }
}
```

CommentService

```java
@Service
public class CommentService {

    private final CommentMapper commentMapper;

    @Autowired
    public CommentService(CommentMapper commentMapper) {
        this.commentMapper = commentMapper;
    }

    // 댓글 입력 결과 응답
    public boolean createComment(CommentDTO commentDto) {
        try {
            int insertResult = commentMapper.createComment(commentDto);
            if (insertResult > 0) {
                return true;
            } else {
                return false;
            }
        } catch (Exception e) {
            e.printStackTrace();
            return false;
        }

    }

    // 게시글에 달린 댓글 리스트 반환 메서드
    public List<CommentDTO> findAllByBoardIdx(int boardIdx) {
        return commentMapper.findAllByBoardIdx(boardIdx);
    }

}
```

CommentMapper.java

```java
package com.example.boardProject.comment;

import java.util.List;

@Mapper
public interface CommentMapper {
    // 댓글 생성
    int createComment(CommentDTO dto);

    // 게시판 번호에 연결된 댓글 리스트 조회
    List<CommentDTO> findAllByBoardIdx(int boardIdx);

}
```

CommentMapper.xml

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.example.boardProject.comment.CommentMapper">


    <!-- 새 댓글 추가 -->
    <insert id="createComment" parameterType="commentdto" useGeneratedKeys="true" keyProperty="idx">
        INSERT INTO comment(board_idx, user_name, content)
        VALUES(#{boardIdx}, #{userName}, #{content})
    </insert>

    <!-- 게시글 ID로 댓글 조회 -->
    <select id="findAllByBoardIdx" resultType="commentdto" parameterType="int">
        SELECT
            idx,
            board_idx,
            user_name,
            content,
            DATE_FORMAT(created_at, '%Y-%m-%d %H:%i:%s') AS created_at
        FROM comment
        WHERE board_idx = #{boardIdx}
        ORDER BY created_at DESC
    </select>

</mapper>
```

BoardController

```java
// 상세보기 페이지
@GetMapping("/detail/{idx}") // /board/detail/
public String detailPage(@PathVariable int idx, Model model, RedirectAttributes redirectAttributes) {
    BoardDTO board = boardService.findBoardById(idx); // null 값이 있는거
    if (board == null) {
        redirectAttributes.addFlashAttribute("error", "해당 게시글 정보가 없습니다.");
        return "redirect:/board/list";
    }
    // 해당 게시글에 연결된 댓글 리스트 조회
    List<CommentDTO> commentList = commentService.findAllByBoardIdx(idx);

    model.addAttribute("board", board);
    model.addAttribute("commentList", commentList);
    return "board/detail";

}
```