

안내서: Spring Boot, Spring Data JDBC, Spring MVC 를 이용한 게시판 만들기

- 강경미 (carami@nate.com), 김성박 (urstory@gmail.com)

개요

본 프로젝트에서는 학생들이 Spring Boot, Spring Data JDBC, 그리고 Spring MVC를 활용하여 웹 애플리케이션을 개발합니다.

프로젝트 요구 사항

- 글 등록
 - 이름, 제목, 암호, 본문을 입력
 - 등록일, ID는 자동으로 저장
- 글 목록 보기
 - 최신 글부터 보여짐
 - ID, 제목, 이름, 등록일(YYYY/MM/DD) 형식으로 목록이 보여짐
 - 페이지징 처리 필요
- 글 상세 조회
 - 암호는 보여지면 안됨
 - 글 등록일은 YYYY/MM/DD hh24:mi 형식으로 보여짐

- 수정
 - 이름, 제목, 본문을 수정
 - 암호는 글 등록시 입력했던 암호를 입력해야함
 - 수정일은 자동으로 저장
- 삭제
 - 암호는 글 등록시 입력했던 암호를 입력해야함

DB 테이블 설계 및 Sample Data

0. 테이블 삭제

```
drop table board;
```

1. Board 테이블

- 사용자 정보를 저장합니다.

```
CREATE TABLE board (  
  id BIGINT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(100) NOT NULL,  
  title VARCHAR(255) NOT NULL,  
  password VARCHAR(255) NOT NULL, -- 암호는 해싱하여 저장하는 것이 좋습니다  
  content TEXT NOT NULL,  
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP, -- 등록일  
  updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP -- 수정일  
);
```

예제 데이터 삽입

```
INSERT INTO board (name, title, password, content) VALUES
('김민수', '첫 번째 글입니다!', 'password123', '이것은 첫 번째 게시글의 내용입니다.'),
('이지은', '봄철 원예 팁', 'password123', '봄철 정원 가꾸기에 유용한 팁을 공유합니다.'),
('박영희', '올해의 여행지 추천 10곳', 'password123', '올해 방문하기 좋은 여행지 10곳을 소개합니다.'),
('최준호', 'SQL 이해하기', 'password123', 'SQL과 그 기능들에 대해 깊이 있게 알아봅시다.'),
('황소연', '최고의 코딩 습관', 'password123', '모든 개발자가 따라야 할 코딩 습관에 대해 알아봅시다.'),
('백은영', '사진 촬영 기초', 'password123', '초보자를 위한 사진 촬영 가이드입니다.'),
('윤서준', '기술의 미래', 'password123', '기술과 혁신의 미래에 대한 예측을 해봅시다.'),
('한지민', '예산 내에서 건강하게 먹기', 'password123', '예산을 깨지 않고 건강하게 먹는 방법을 공유합니다.'),
('정태웅', '초보자를 위한 운동 루틴', 'password123', '운동을 막 시작한 사람들을 위한 효과적인 운동 루틴을 소개합니다.'),
('김서영', '지역 뉴스 업데이트', 'password123', '최신 지역 뉴스를 업데이트합니다.'),
('남주혁', '새로운 영화 개봉', 'password123', '이번 주말 개봉하는 새로운 영화들을 확인해보세요.');
```


페이지 별로 데이터 읽어오기

```
select id, name, title, password, content, created_at, updated_at from board order by id desc limit 0,5;
```

1. SELECT 절:

- `id`, `name`, `title`, `password`, `content`, `created_at`, `updated_at` : 이 쿼리에서는 게시판 테이블의 `id`, `name`, `title`, `password`, `content`, `created_at` (생성일시), `updated_at` (수정일시) 등 7개의 컬럼을 선택하고 있습니다. 이를 통해 각 게시글의 주요 정보와 함께 생성 및 수정 날짜를 확인할 수 있습니다.

2. FROM 절:

- `board` : 데이터를 검색할 테이블의 이름입니다. 이 경우, `board` 테이블에서 데이터를 추출합니다.

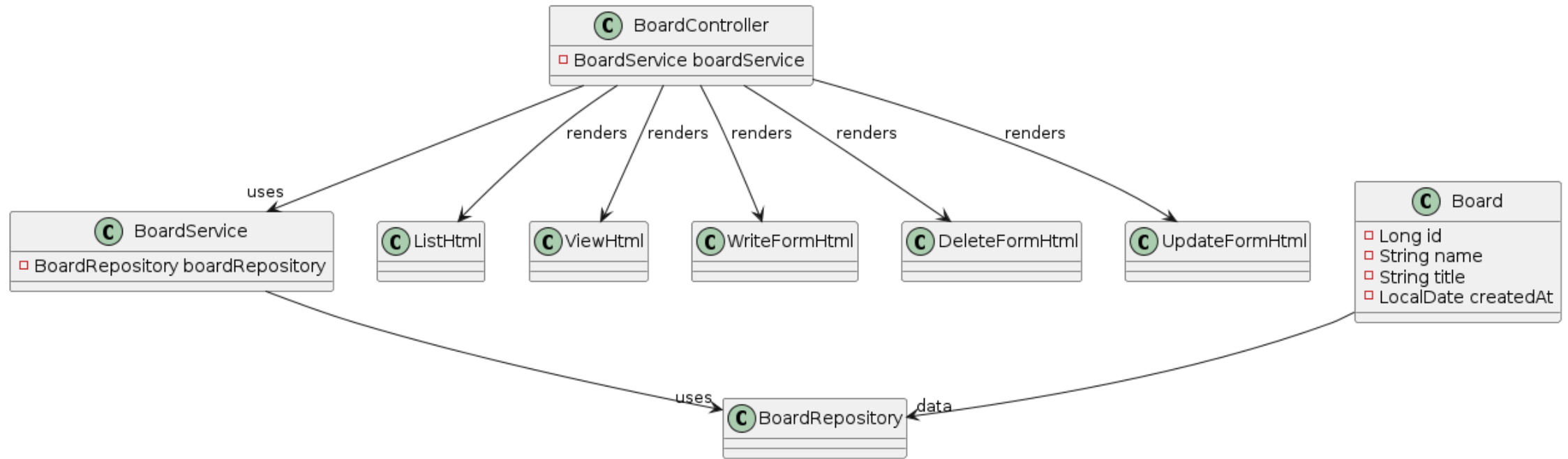
3. ORDER BY 절:

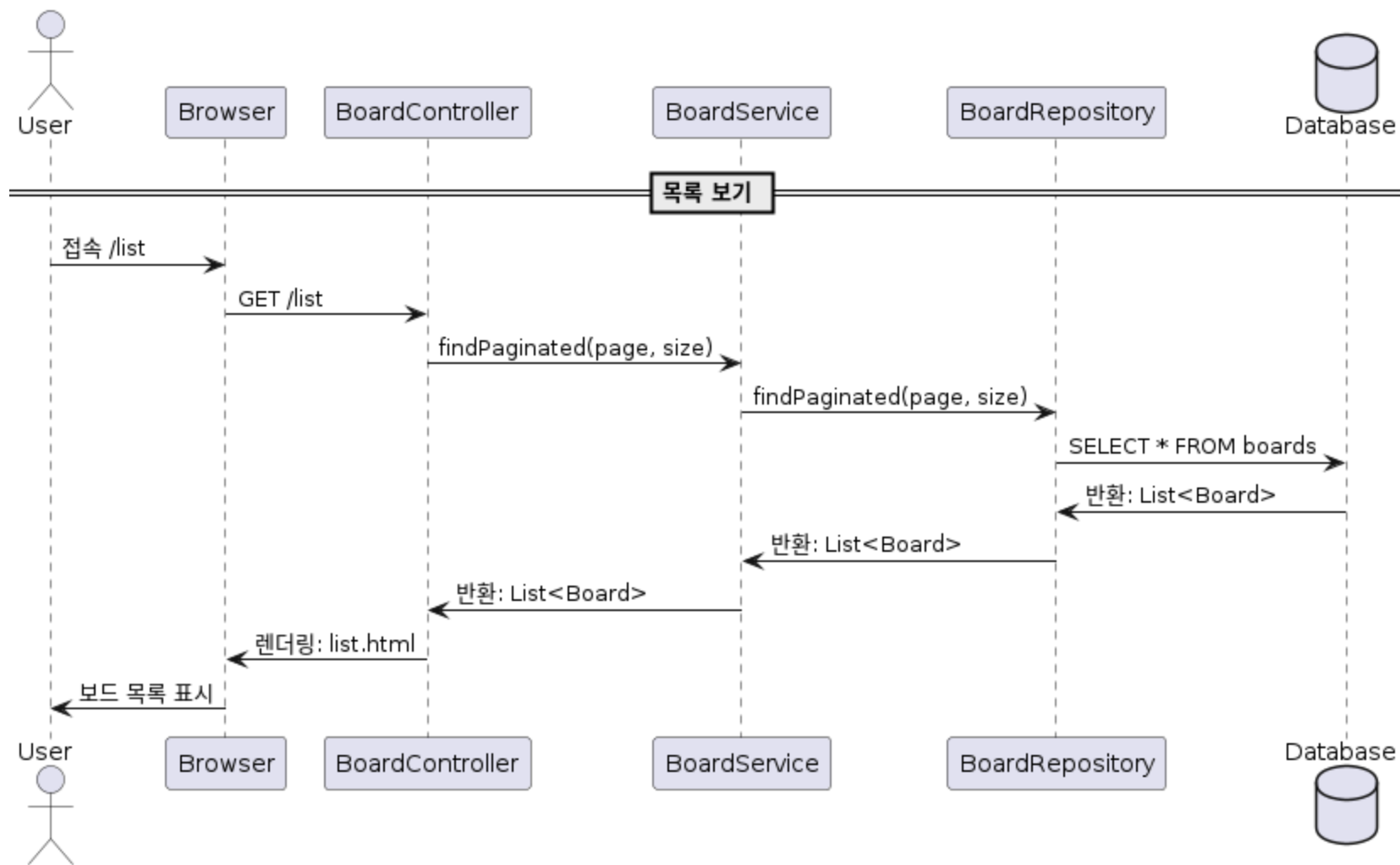
- `ORDER BY id DESC` : 게시글을 `id` 컬럼을 기준으로 내림차순으로 정렬합니다. `id` 값이 큰 게시물일수록 더 최근에 작성된 게시물이기 때문에, 이 정렬을 통해 최신 게시물을 상위에 위치시킬 수 있습니다.

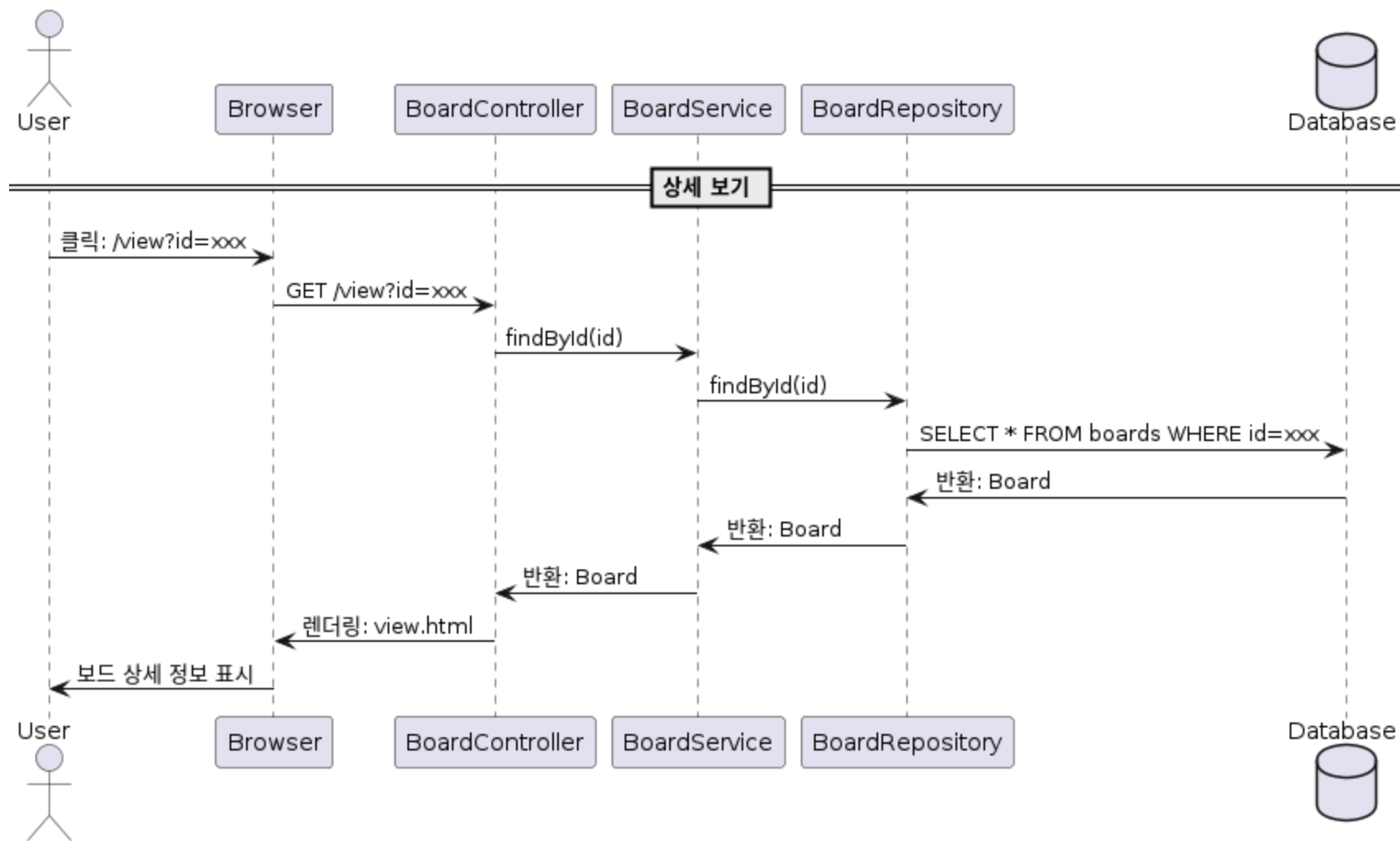
4. LIMIT 절:

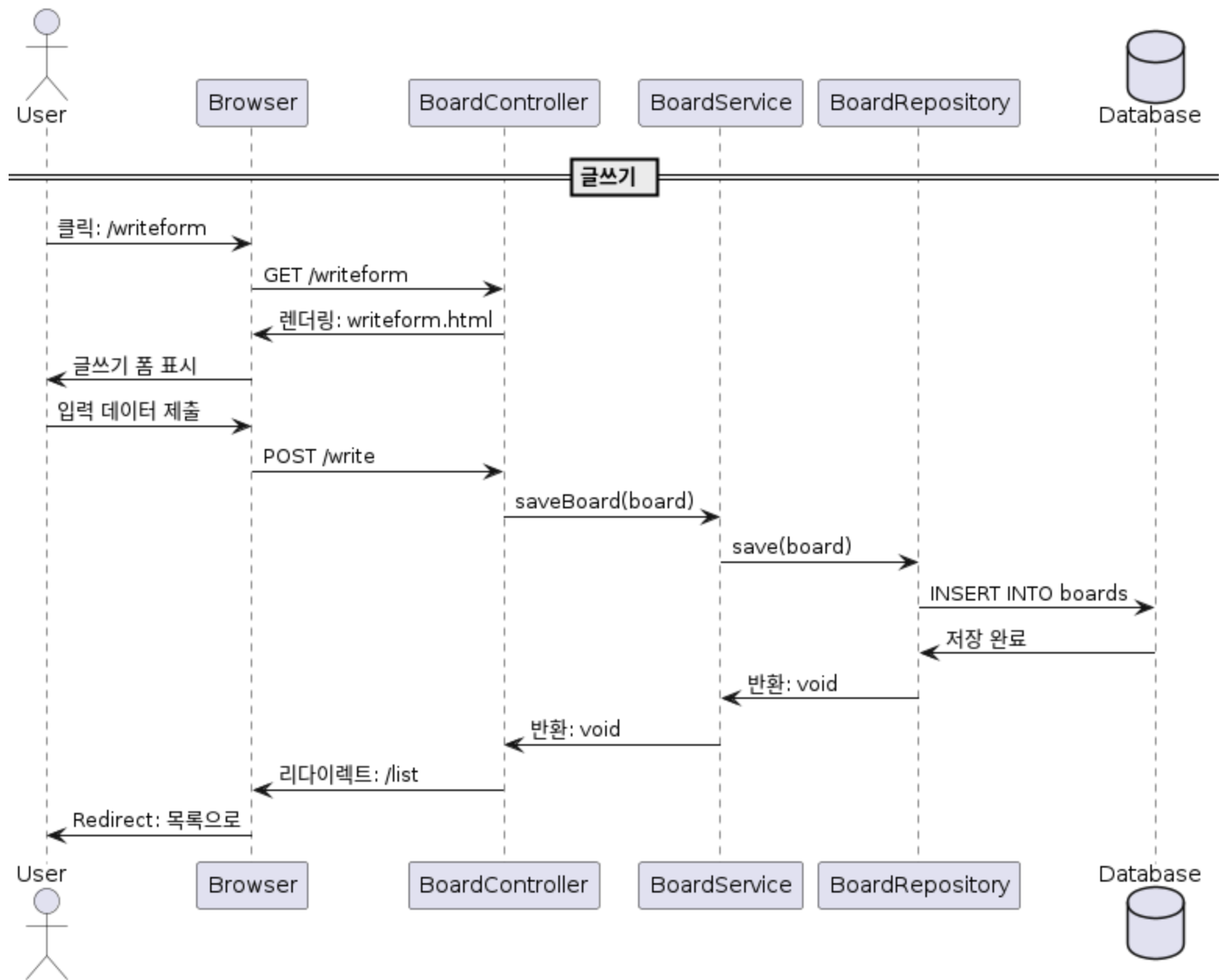
- `LIMIT 0, 5` : 이 부분은 검색 결과의 양을 제한하는 데 사용됩니다. `LIMIT 0, 5` 는 결과 집합의 첫 번째 행부터 시작하여 5개의 행을 반환하라는 의미입니다. 즉, 가장 최근에 등록된 게시글 5개를 가져옵니다.

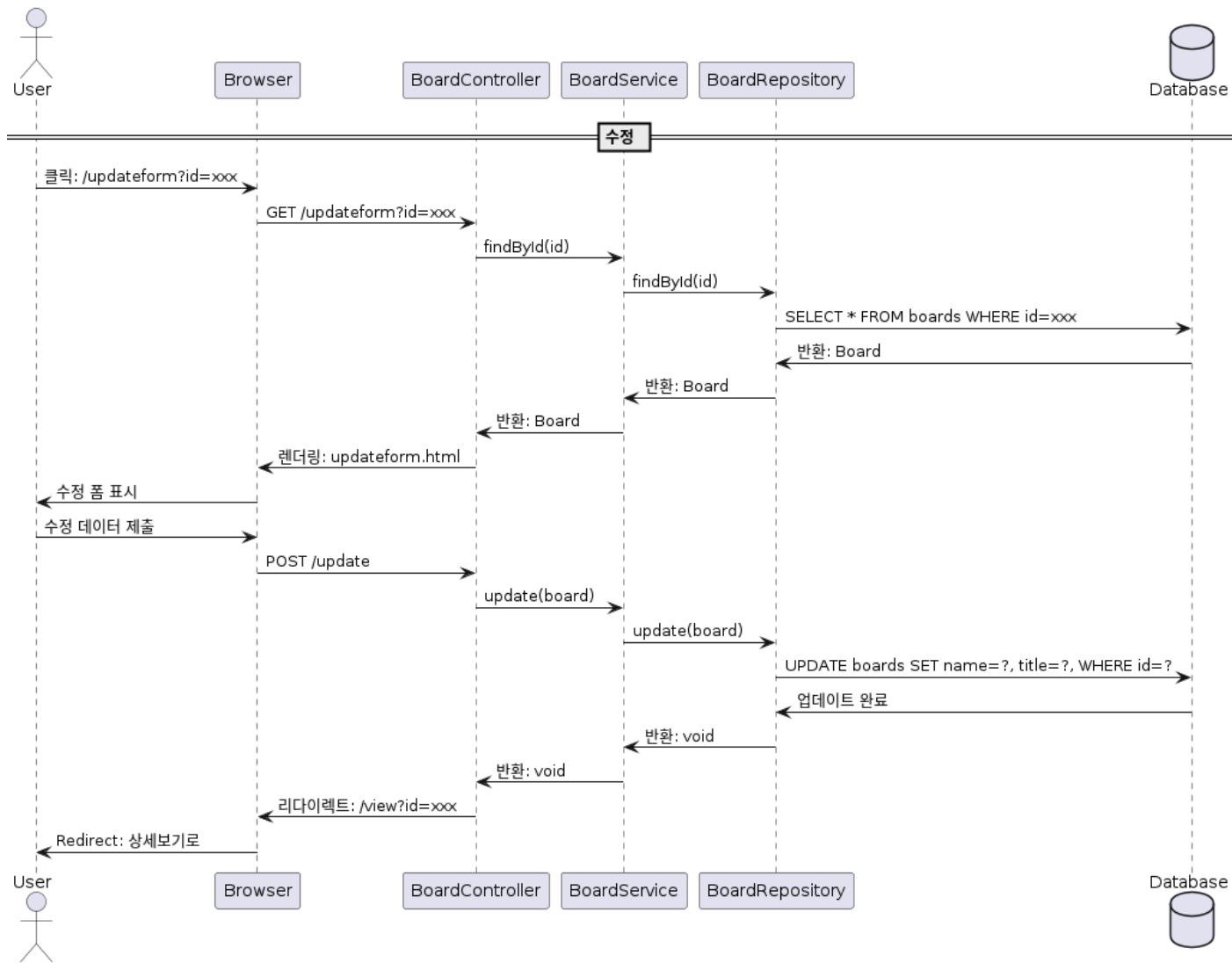
요청/응답 처리 클래스 & 시퀀스 다이어그램

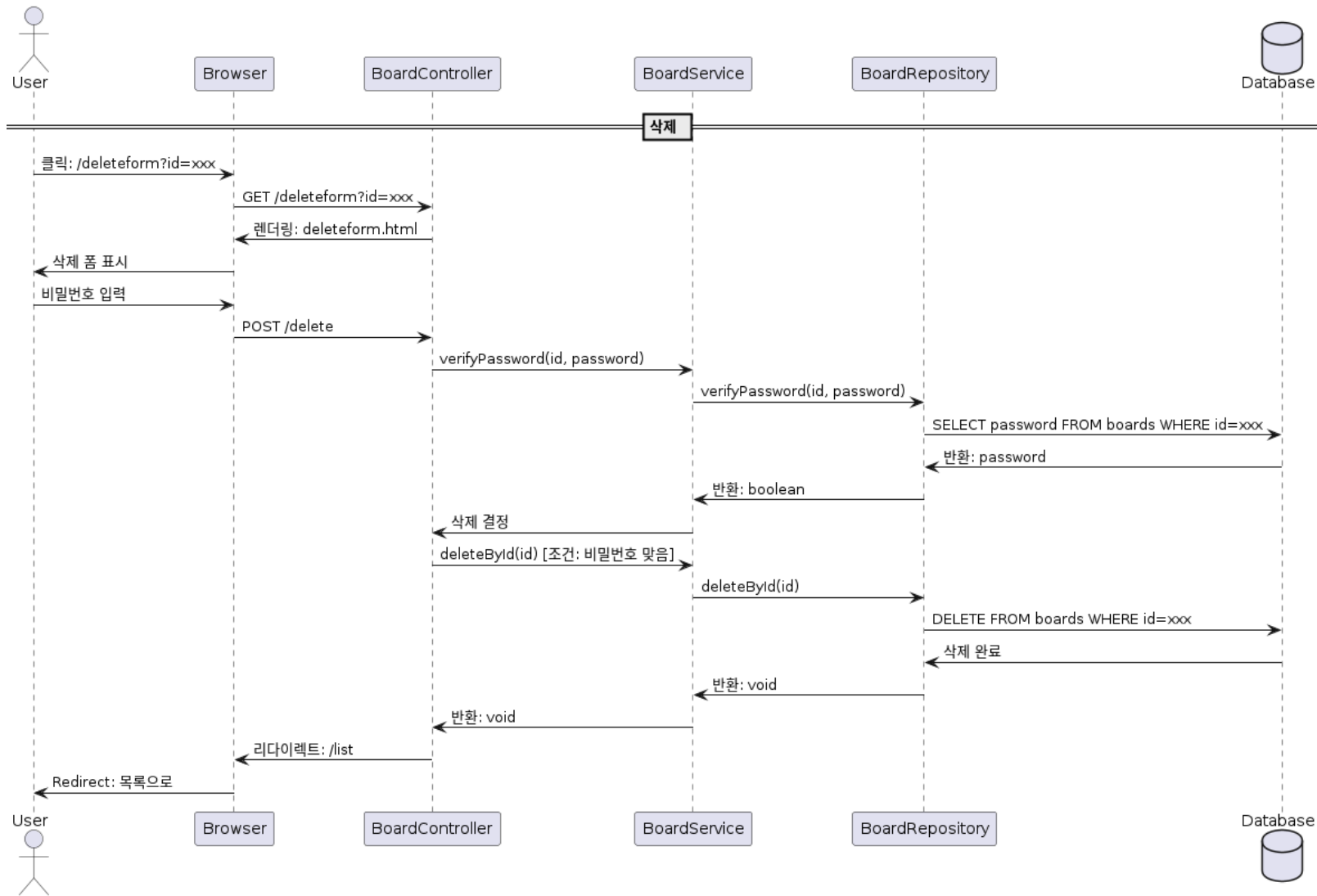












URL 구조

1. 게시물 목록 보기 (/list)

- URL: /list , /list?page=2
- 기능:
 - 게시물 목록을 페이지별로 보여줍니다.
 - page 파라미터가 없으면 기본적으로 1페이지를 보여줍니다.
 - 각 페이지는 최신 글부터 보여지며,페이징 처리가 적용되어 있습니다.
 - 하단에는 페이지 네비게이터가 있어 다른 페이지로 쉽게 이동할 수 있습니다.
 - 각 게시물은 ID, 제목, 이름, 등록일(YYYY/MM/DD 형식)로 목록이 구성됩니다.

Board List

ID	Name	Title	Created At
11	남주혁	새로운 영화 개봉	2024/05/11
10	김서영	지역 뉴스 업데이트	2024/05/11
9	정태웅	초보자를 위한 운동 루틴	2024/05/11
8	한지민	예산 내에서 건강하게 먹기	2024/05/11
7	윤서준	기술의 미래	2024/05/11

[1](#) [2](#) [3](#)

[글쓰기](#)

Board List

ID	Name	Title	Created At
6	백은영	사진 촬영 기초	2024/05/11
5	황소연	최고의 코딩 습관	2024/05/11
4	최준호	SQL 이해하기	2024/05/11
3	박영희	올해의 여행지 추천 10곳	2024/05/11
2	이지은	봄철 원예 팁	2024/05/11

1 2 3

[글쓰기](#)

2. 게시물 상세 조회 (/view?id=아이디)

- URL: /view?id=아이디
- 기능:
 - 특정 게시글의 상세 내용을 보여줍니다.
 - 삭제와 수정 링크를 제공하여 해당 기능을 수행할 수 있는 페이지로 이동할 수 있습니다.
 - 게시글의 등록일은 YYYY/MM/DD hh24:mi 형식으로 표시됩니다.
 - 게시글의 암호는 보여지지 않습니다.



localhost:8080/view?id=11&page=1



새로운 영화 개봉

Name: 남주혁

Content: 이번 주말 개봉하는 새로운 영화들을 확인해보세요.

Created At: 2024/05/11 12:34

[List](#) [Edit](#) [Delete](#)

3. 게시물 등록 폼 (/writeform)

- URL: /writeform
- 기능:
 - 특정 게시글을 쓰기위한 폼을 제공합니다.
 - 사용자는 이름, 제목, 내용, 암호를 입력하고, 확인 버튼을 클릭하여 등록을 요청합니다.
 - 모든 내용이 잘 입력되어 있을 경우 /write 로 요청을 보내 등록 처리 후 /list 로 리다이렉트됩니다.

Create a New Board

Name:

Title:

Content:

Password:

Create

4. 게시물 삭제 폼 (/deleteform?id=아이디)

- URL: /deleteform?id=아이디
- 기능:
 - 특정 게시글을 삭제하기 위한 폼을 제공합니다.
 - 사용자는 암호를 입력하고, 확인 버튼을 클릭하여 삭제를 요청합니다.
 - 올바른 암호 입력 시, /delete 로 요청을 보내 삭제 처리 후 /list 로 리다이렉트됩니다.

Delete Board

Password:

....

Delete

5. 게시물 수정 폼 (/updateform?id=아이디)

- URL: /updateform?id=아이디
- 기능:
 - 특정 게시글을 수정하기 위한 폼을 제공합니다.
 - 이름, 제목, 본문, 암호 필드를 포함하며, 사용자는 이를 수정할 수 있습니다.
 - 확인 버튼을 클릭하면 /update 로 수정 요청을 보내고, 수정이 완료되면 해당 게시글의 상세 페이지(/view?id=아이디)로 리다이렉트됩니다.

Update Board

Name:

Title:

Content:

Password (for verification):

Update

개발 단계

1. 환경 설정

- Spring Boot 프로젝트 생성
- 필요한 의존성 추가 (Spring Web, Spring Data JDBC 등)

2. 데이터베이스 구성

- SQL 스크립트를 사용하여 Users, Posts, Comments 테이블 생성 및 초기 데이터 삽입

3. 도메인 모델 생성

- 각 테이블에 대응하는 Java 클래스(도메인 모델) 생성

4. DAO 개발

- Spring Data JDBC를 이용한 Repository 인터페이스 구현

5. 서비스 계층 구현

- 비즈니스 로직을 수행하는 서비스 클래스 구현

6. 컨트롤러 구현

- 컨트롤러 클래스 구현
- thymeleaf템플릿 작성

