Spring Boot

소프트웨어융합공학과
웹서버 프로그래밍
허우행

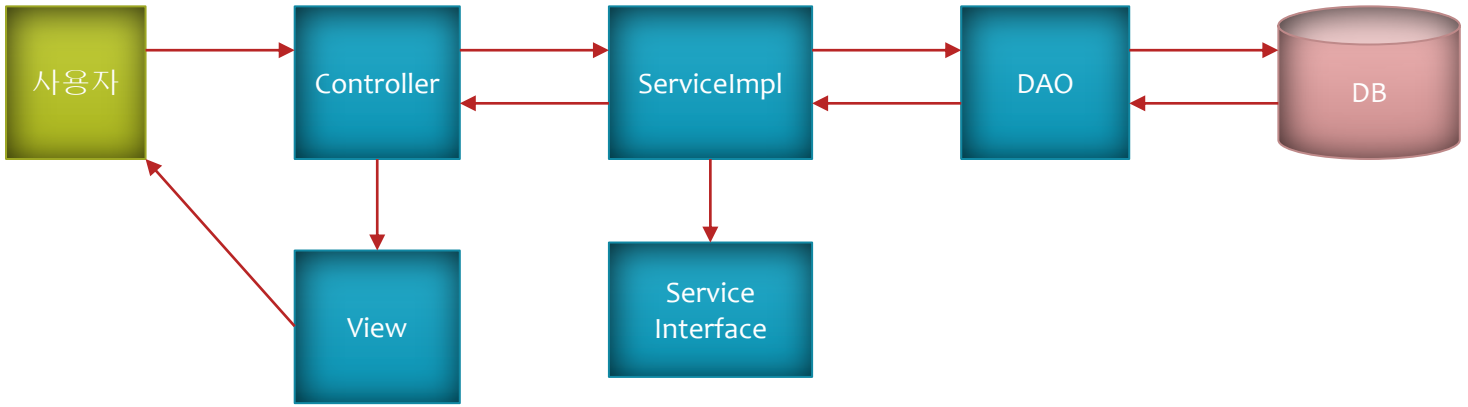
게시판 테이블 생성

- HeideSQL 이나 Workbench 를 이용해서 직접 생성하는 방법과 스크립트를 로딩하는 방법이 있다.
- SQL 스크립트

```
CREATE TABLE `t_board_2` (
`board_idx` INT(11) NOT NULL AUTO_INCREMENT COMMENT '글번호',
`title` VARCHAR(300) NOT NULL COMMENT '제목' COLLATE 'utf8_general_ci',
`contents` MEDIUMTEXT NULL DEFAULT NULL COMMENT '내용' COLLATE 'utf8_general_ci',
`hit_cnt` INT(11) NULL DEFAULT NULL COMMENT '조회수',
`created_datetime` DATETIME NULL DEFAULT NULL COMMENT '작성시간',
`creator_id` VARCHAR(50) NULL DEFAULT NULL COMMENT '생성자' COLLATE 'utf8_general_ci',
`updated_datetime` DATETIME NULL DEFAULT NULL COMMENT '수정시간',
`updater_id` VARCHAR(50) NULL DEFAULT NULL COMMENT '수정자' COLLATE 'utf8_general_ci',
`deleted_yn` CHAR(1) NULL DEFAULT NULL COMMENT '삭제여부' COLLATE 'utf8_general_ci',
PRIMARY KEY (`board_idx`) USING BTREE,
INDEX `title` (`title`) USING BTREE
)
COMMENT='게시판'
COLLATE= 'utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=1
```

Spring Boot 웹(MVC)

- MVC 구조
  - Model : 어플리케이션의 정보, 즉 데이터(DB)를 의미합니다.
  - View : 사용자가 보는 화면 또는 결과를 의미합니다.
  - Controller : 사용자가 웹브라우저를 통해서 어떠한 요청을 하면 그 요청을 처리할 컨트롤러를 호출하게 됩니다. 컨트롤러는 사용자의 요청을 처리하기 위한 비즈니스 로직을 호출하고 그 결과값을 사용자에게 전달해 주는 역할을 합니다.
  - Service : 사용자의 요청을 처리하기 위한 비즈니스 로직이 수행됩니다. 일반적으로 서비스 영역은 서비스 인터페이스와 이 인터페이스의 구현체로 나뉩니다.
  - DAO : Data Access Object 의 약자로 데이터베이스에 접속해서 비즈니스 로직 실행에 필요한 쿼리를 호출합니다.
  - DB : 데이터베이스를 의미합니다. 데이터베이스에는 애플리케이션에서 발생한 모든 정보가 저장되어 있습니다.



스타일 시트 추가

/board/src/main/resources/static/css/style.css

```
@CHARSET "UTF-8";

@import url(http://fonts.googleapis.com/earlyaccess/nanumgothic.css);
@import url(http://cdn.jsdelivr.net/font-nanum/1.0/nanumbarungothic/nanumbarungothic.css);

html{overflow:scorll;}
html, body, div, h1, h2, a, form, table, caption, thead, tbody, tr, th, td, submit {
   margin:0; outline:0; border:0; padding:0; font-size:100%; vertical-align:baseline; background:transparent;
}
body {
   font-size:0.875em; line-height:1.5; color:#666; -webkit-text-size-adjust:none; min-width:320px;
   font-family:'NanumGothic','나눔고딕',dotum, "Helvetica Neue", Helvetica, Verdana, Arial, Sans-Serief;
}

h1, h2, h3 {font-size: 1.5em;}
p{margin:0; padding:0;}
ul{margin:0;}
a:link, a:visited {text-decoration:none; color: #656565;}
input{vertical-align:middle;}
input:focus {outline:0;}
caption {display:none; width:0; height:0; margin-top:-1px; overflow:hidden; visibility:hidden; font-size:0; line-height:0;}
```

스타일 시트 추가

```
.container {max-width:1024px; margin:30px auto;}
.board_list {width:100%; border-top:2px solid #252525; border-bottom:1px solid #ccc; margin:15px 0; border-collapse: collapse;}
.board_list thead th:first-child {background-image:none;}
.board_list thead th {border-bottom:1px solid #ccc; padding:13px 0; color:#3b3a3a; text-align: center; vertical-align:middle;}
.board_list tbody td {border-top:1px solid #ccc; padding:13px 0; text-align:center; vertical-align:middle;}
.board_list tbody tr:first-child td {border:none;}
.board_list tbody tr:hover{background:#ffff99;}
.board_list tbody td.title {text-align:left; padding-left:20px;}
.board_list tbody td a {display:inline-block}
.board_detail {width:100%; border-top:2px solid #252525; border-bottom:1px solid #ccc; border-collapse:collapse;}
.board_detail tbody input {width:100%;}
.board_detail tbody th {text-align:left; background:#f7f7f7; color:#3b3a3a; vertical-align:middle; text-align: center;}
.board_detail tbody th, .board_detail tbody td {padding:10px 15px; border-bottom:1px solid #ccc;}
.board_detail tbody textarea {width:100%; min-height:170px}
.btn {margin:5px; padding:5px 11px; color:#fff !important; display:inline-block; background-color:#7D7F82; vertical-align:middle; border-radius:0 !important;
cursor:pointer; border:none;}
.btn:hover {background: #6b9ab8;}
.file_list a {display:inherit !important;}
```

마이바티스 설정

/board/src/main/resources/application.properties ← 컬럼명을 카멜 표기법으로 변경

```
Mybatis.configuration.map-underscore-to-camel-case=true
```

/board/src/main/java/kr/ac/inha/configuration/DatabaseConfiguration.java

```
@Bean
@ConfigurationProperties(prefix="mybatis.configuration")
public org.apache.ibatis.session.Configuration mybatisConfig(){
        return new org.apache.ibatis.session.Configuration();
}

@Bean
public SqlSessionFactory sqlSessionFactory(DataSource dataSource) throws Exception {
    SqlSessionFactoryBean sqlSessionFactoryBean = new SqlSessionFactoryBean();
    sqlSessionFactoryBean.setDataSource(dataSource);
    sqlSessionFactoryBean.setMapperLocations(applicationContext.getResources("classpath:/mapper/**/sql-*.xml"));
    sqlSessionFactoryBean.setConfiguration(mybatisConfig());

    return sqlSessionFactoryBean.getObject();
}
```

← **application.properties** 읽어 오는 부분

← 추가(환경설정 적용)

DTO 만들기

- Dto 의 변수는 기본적으로 카멜 표기법을 사용한다.

/board/src/main/java/kr/ac/inha/board/board/dto/BoardDto.java

```
package kr.ac.inha.board.board.dto;

import lombok.Data;

@Data
public class BoardDto {
    private int boardIdx;
    private String title;
    private String contents;
    private int hitCnt;
    private String createdDatetime;
    private String creatorId;
    private String updateDatetime;
    private String updaterId;
    private String deletedYn;
}
```

테이블당 하나씩 생성

변수명은 카멜 표기법 사용

1. 카멜 표기법

- "camelCase"
- "단봉낙타" 표기법[1][2]
- 각 단어의 첫문자를 대문자로 표기하고 붙여쓰되, 맨처음 문자는 소문자로 표기함
- 띄어쓰기 대신 대문자로 단어를 구분하는 표기 방식
- 예시: backgroundColor, typeName, iPhone

2. 파스칼 표기법

- "PascalCase"
- 첫 단어를 대문자로 시작하는 표기법
- 예시: BackgroundColor, TypeName, PowerPoint

Controller 작성

/board/src/main/java/kr/ac/inha/board/board/controller/BoardController.java

```
package kr.ac.inha.board.board.controller;

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

import kr.ac.inha.board.board.dto.BoardDto;

@Controller
public class BoardController {
	@Autowired
	private BoardService boardService;

	@RequestMapping("/board/openBoardList.do")
	public ModelAndView openBoardList() throws Exception {
		ModelAndView mv = new ModelAndView("/board/boardList");

		List<BoardDto> list = boardService.selectBoardList();
		mv.addObject("list", list);

		return mv;
	}
}
```

<CTRL>+<SHIFT>+"O" 하면 import 문 자동생성됨

Service 작성

/board/src/main/java/kr/ac/inha/board/board/service/BoardService.java

```
package com.qhedge.online.board.board.service;
…
public interface BoardService {
List<BoardDto> selectBoardList() throws Exception;
}
```

/board/src/main/java/kr/ac/inha/board/board/service/BoardServiceImpl.java

```
package com.qhedge.online.board.board.service;
…
@Service
public class BoardServiceImpl implements BoardService{
    @Autowired
    private BoardMapper boardMapper;

    @Override
    public List<BoardDto> selectBoardList() throws Exception {
        return boardMapper.selectBoardList();
    }
}
```

Mapper 작성

/board/src/main/java/kr/ac/inha/board/board/mapper/BoardMapper.java

```
package com.qhedge.online.board.board.mapper;

import java.util.List;
import org.apache.ibatis.annotations.Mapper;
import com.qhedge.online.board.board.dto.BoardDto;

@Mapper
public interface BoardMapper {
        List<BoardDto> selectBoardList() throws Exception;
}
```

SQL 작성

/board/src/main/resources/mapper/sql-board.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="kr.ac.inha.board.board.mapper.BoardMapper">
<select id="selectBoardList" resultType="kr.ac.inha.board.board.dto.BoardDto">
<![CDATA[
    SELECT
        board_idx,
        title,
        hit_cnt,
        DATE_FORMAT(created_datetime, '%Y-%m-%d %H:%i:%s') AS created_datetime
    FROM
        t_board
    WHERE
        deleted_yn = 'N'
    ORDER BY board_idx DESC
]]>
</select>
</mapper>
```

구분기호	역할	구분기호	역할
%Y	4자리 년도	%m	숫자 월 (두자리)
%y	2자리 년도	%c	숫자 월(한자리는 한자리)
%M	긴 월(영문)	%d	일자 (두자리)
%b	짧은 월(영문)	%e	일자(한자리는 한자리)
%W	긴 요일 이름(영문)	%l	시간 (12시간)
%a	짧은 요일 이름(영문)	%H	시간(24시간)
%i	분	%r	hh:mm:ss AM,PM
%T	hh:mm:SS	%S	초

뷰 작성

/board/src/main/resources/templates/board/boardList.html

```
<!DOCTYPE html>
<html lang="ko" xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<title>게시판</title>
<link rel="stylesheet" th:href="@{/css/style.css}"/>
</head>
<body>
    <div class="container">
    <h2>게시글 목록</h2>
    <table class="board_list">
        <colgroup>
            <col width="15%"/>
            <col width="*"/>
            <col width="15%"/>
            <col width="20%"/>
        </colgroup>
        <thead>
            <tr>
                <th scope="col">글번호</th>
                <th scope="col">제목</th>
                <th scope="col">조회수</th>
                <th scope="col">작성일</th>
            </tr>
        </thead>
```

뷰 작성

```
<tbody>
        <tr th:if="${#lists.size(list)} > 0" th:each="list : ${list}">
           <td th:text="${list.boardIdx}"></td>
           <td class="title"><a href="/board/openBoardDetail.do?boardIdx=" th:attrappend="href=${list.boardIdx}“
            th:text="${list.title}"></a></td>
           <td th:text="${list.hitCnt}"></td>
           <td th:text="${list.createdDatetime}"></td>
        </tr>
        <tr th:unless="${#lists.size(list)} > 0">
           <td colspan="4">조회된 결과가 없습니다.</td>
        </tr>
        </tbody>
    </table>
    <p align="right">
    <a href="/board/openBoardWrite.do" class="btn">글 쓰기</a>
    </p>
    </div>
</body>
</html>
```