

A Glory Day 1

Free Bootstrap Admin Template - AdminLTE.IO

<https://adminlte.io/>

✓ DB 설정

✓ Project 생성

✓ MyBatis 설정

✓ Board

✓ Notice

✓ Member

아파트 관리 사무소 페이지

| 정해진 회원만 접근 가능, 요청 / 접수 /처리 홈페이지

DB

Table

glorynotice

Column	Datatype	PK	NN	UQ	B...	UN	ZF	AI	G	Default / Expression
nid	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL				
ntitle	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
ncontent	LONGTEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
ncate	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
nwriter	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'admin'
ndate	TIMESTAMP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
<click to edit>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

gloryboard

Column	Datatype	PK	NN	UQ	B...	UN	ZF	AI	G	Default / Expression
bid	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
bpid	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
btitle	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
bcontent	LONGTEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
bcate	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
bstatus	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'1'
bwriter	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
bdate	TIMESTAMP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
<click to edit>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

glorymember

스프링 시큐리티에 맞게 생성 필요

Column	Datatype	PK	NN	UQ	B...	UN	ZF	AI	G	Default / Expression
mid	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
enabled	TINYINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'1'
username	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
password	VARCHAR(200)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
authority	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'ROLE_USER'
mname	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
memail	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
mtel	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
mdate	TIMESTAMP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP
<click to edit>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

```

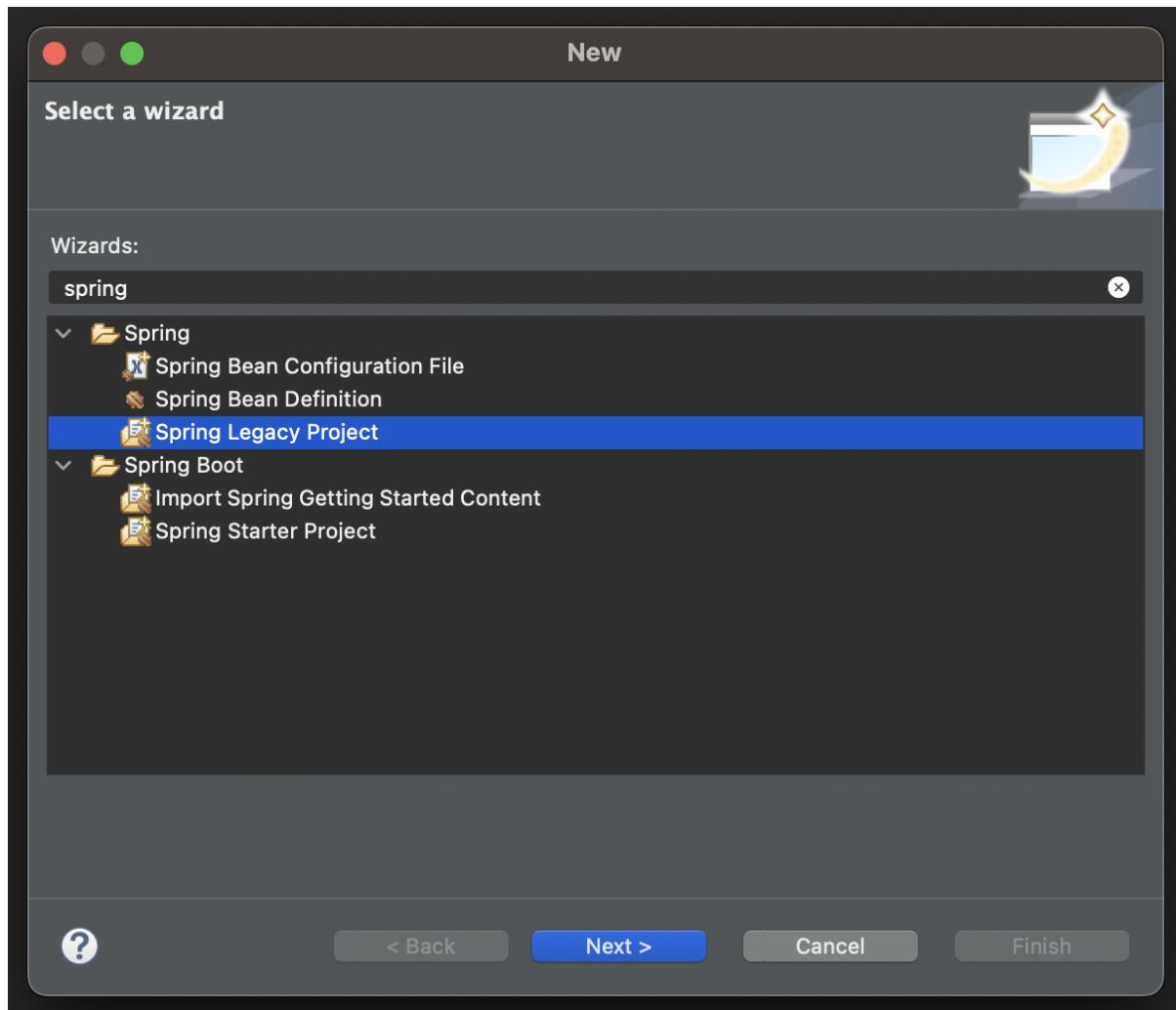
CREATE TABLE `glorynotice` (
  `nid` int NOT NULL AUTO_INCREMENT,
  `ntitle` varchar(45) DEFAULT NULL,
  `ncontent` longtext,
  `ncate` varchar(45) DEFAULT NULL,
  `nwriter` varchar(45) DEFAULT 'admin',
  `ndate` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`nid`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

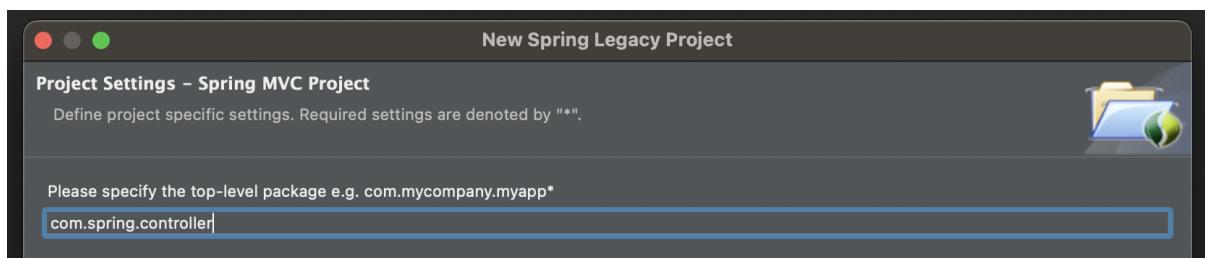
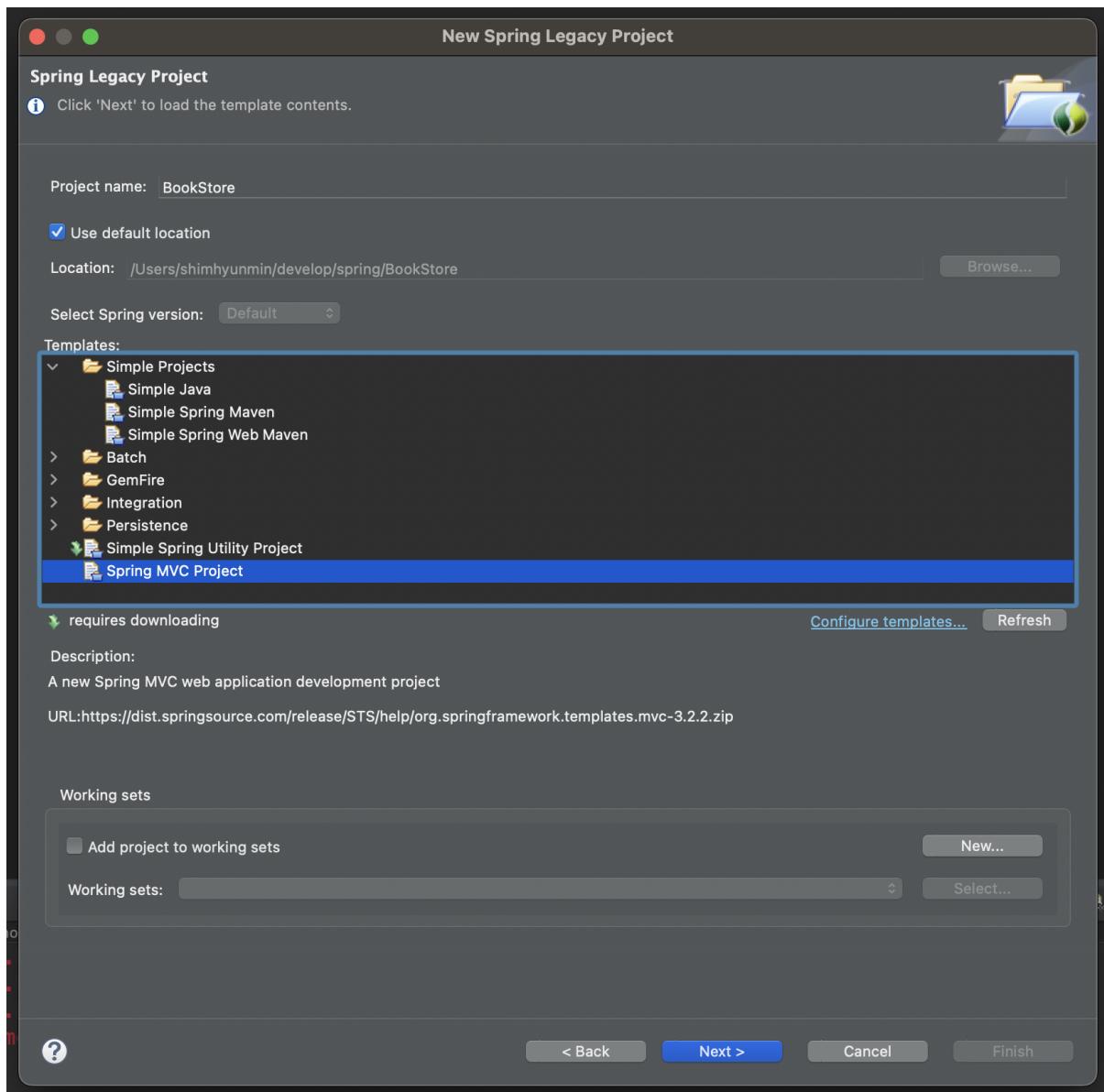
CREATE TABLE `gloryboard` (
  `bid` int NOT NULL AUTO_INCREMENT,
  `bpid` varchar(45) DEFAULT '0',
  `btitle` varchar(45) DEFAULT NULL,
  `bcontent` longtext,
  `bcate` varchar(45) DEFAULT NULL,
  `bstatus` int DEFAULT '1',
  `bwriter` varchar(45) DEFAULT NULL,
  `bdate` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`bid`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `glorymember` (
  `mid` int NOT NULL AUTO_INCREMENT,
  `enabled` tinyint DEFAULT '1',
  `username` varchar(45) NOT NULL,
  `password` varchar(200) DEFAULT NULL,
  `authority` varchar(45) DEFAULT 'ROLE_USER',
  `mname` varchar(45) DEFAULT NULL,
  `memail` varchar(45) DEFAULT NULL,
  `mtel` varchar(45) DEFAULT NULL,
  `mdate` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`mid`, `username`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

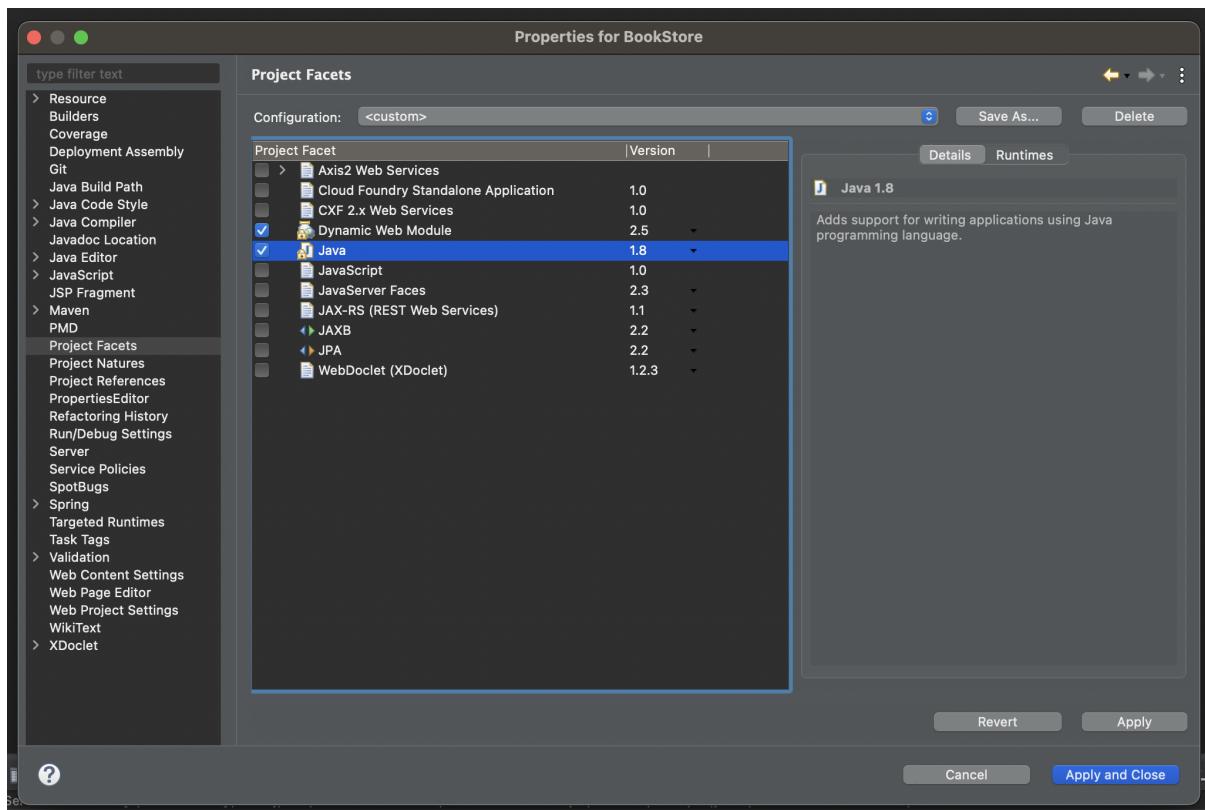
```

Project 생성





자바 버전 변경



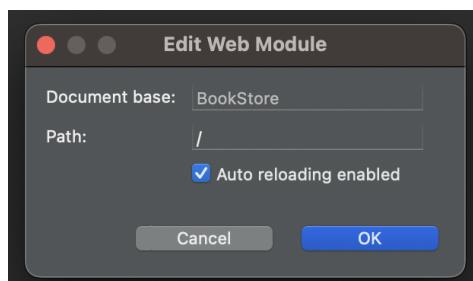
스프링 버전 변경

pom.xml

```
<org.springframework-version>5.3.19</org.springframework-version>
```

웹 경로 설정

server



package 경로 설정

servelt-context.xml

```
<context:component-scan base-package="com.aglory.*" />
```

한글처리

web.xml

```
<filter>
    <filter-name>encodingFilter</filter-name>
    <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
    <init-param>
        <param-name>encoding</param-name>
        <param-value>UTF-8</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>encodingFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

MyBatis 설정

pom.xml

```
<!-- mybatis -->
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
    <version>3.5.6</version>
</dependency>

<!-- mybatis-spring -->
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis-spring</artifactId>
    <version>2.0.4</version>
</dependency>

<!-- spring-jdbc -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>5.3.19</version>
</dependency>

<!-- commons-dbc2 -->
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-dbc2</artifactId>
    <version>2.7.0</version>
</dependency>

<!-- log4jdbc-log4j2-jdbc4 -->
<dependency>
    <groupId>org.bgee.log4jdbc-log4j2</groupId>
    <artifactId>log4jdbc-log4j2-jdbc4</artifactId>
    <version>1.16</version>
</dependency>

<!-- mysql-connector-java -->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.28</version>
</dependency>
```

DB 연결

root-context.xml

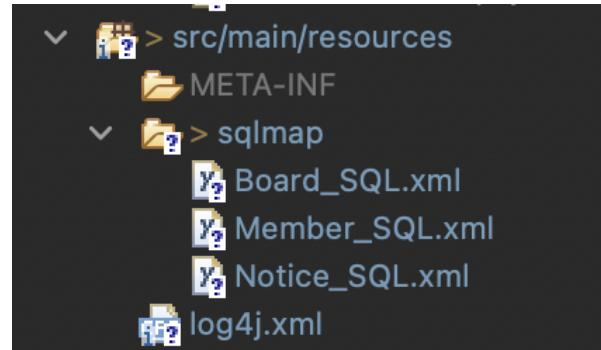
```
<!-- mysql 연결 설정 -->
<bean id="dataSource"
    class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="driverClassName"
        value="com.mysql.cj.jdbc.Driver" />
```

```

<property name="url"
  value="jdbc:mysql://localhost:3306/#?serverTimezone=UTC" />
<property name="username" value="#" />
<property name="password" value="#" />
</bean>
<bean id="sqlSessionFactory"
  class="org.mybatis.spring.SqlSessionFactoryBean">
  <property name="dataSource" ref="dataSource" />
  <property name="mapperLocations"
    value="classpath:/sqlmap/**/*_SQL.xml" />
</bean>
<bean id="sqlSessionTemplate"
  class="org.mybatis.spring.SqlSessionTemplate">
  <constructor-arg index="0" ref="sqlSessionFactory" />
</bean>

```

Mapper



Board

DTO

```

package com.aglory.board;

public class Board {

    private int bid, bstatus;

    private String bpid, btitle, bcontent, bcate, bwriter, bdate;

    public int getBid() {
        return bid;
    }

    public void setBid(int bid) {
        this.bid = bid;
    }

    public int getBstatus() {
        return bstatus;
    }

    public void setBstatus(int bstatus) {
        this.bstatus = bstatus;
    }

    public String getBpid() {
        return bpid;
    }

    public void setBpid(String bpid) {
        this.bpid = bpid;
    }

    public String getBtitle() {
        return btitle;
    }
}

```

```

public void setBtitle(String btitle) {
    this.btitle = btitle;
}

public String getBcontent() {
    return bcontent;
}

public void setBcontent(String bcontent) {
    this.bcontent = bcontent;
}

public String getBcate() {
    return bcate;
}

public void setBcate(String bcate) {
    this.bcate = bcate;
}

public String getBwriter() {
    return bwriter;
}

public void setBwriter(String bwriter) {
    this.bwriter = bwriter;
}

public String getBdate() {
    return bdate;
}

public void setBdate(String bdate) {
    this.bdate = bdate;
}

public Board() {
}

public Board(int bid, int bstatus, String bpid, String btitle, String bcontent, String bcate, String bwriter, String bdate) {
    this.bid = bid;
    this.bstatus = bstatus;
    this.bpid = bpid;
    this.btitle = btitle;
    this.bcontent = bcontent;
    this.bcate = bcate;
    this.bwriter = bwriter;
    this.bdate = bdate;
}
}

```

Repository

```

//C
void setNewBoard(Board board); //게시물 작성

void replynewBoard(Map map); //댓글 작성

//R
List<Board> getAllBoardList(); //게시물 전체 목록

Board getBoardById(String bid); //게시물 하나

List<Board> getReplyById(String bid); //댓글 전체 목록
-----
@Autowired
SqlSessionTemplate sqlSessionTemplate;

public void setNewBoard(Board board) {
    this.sqlSessionTemplate.insert("board.insert", board);
}

public List<Board> getAllBoardList() {
    return this.sqlSessionTemplate.selectList("board.select_list");
}

public Board getBoardById(String bid) {

```

```

        return this.sqlSessionTemplate.selectOne("board.select_detail", bid);
    }

    public void replynewBoard(Map map) {
        this.sqlSessionTemplate.insert("board.insert_reply", map);
    }

    public List<Board> getReplyById(String bid) {
        return this.sqlSessionTemplate.selectList("board.select_reply", bid);
    }

```

Service

```

void setNewBoard(Board board);

void replynewBoard(Map map);

List<Board> getAllBoardList();

Board getBoardById(String bid);

List<Board> getReplyById(String bid);
-----
@Autowired
BoardRepository boardRepository;

public void setNewBoard(Board board) {
    boardRepository.setNewBoard(board);
}

public List<Board> getAllBoardList() {
    return boardRepository.getAllBoardList();
}

public Board getBoardById(String bid) {
    return boardRepository.getBoardById(bid);
}

public void replynewBoard(Map map) {
    boardRepository.replynewBoard(map);
}

public List<Board> getReplyById(String bid) {
    return boardRepository.getReplyById(bid);
}

```

Controller

```

package com.aglory.board;

import java.util.List;
import java.util.Map;

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.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
@RequestMapping("/board")
public class BoardController {

    @Autowired
    private BoardService boardService;

    // @Autowired
    // private MailService mailService;
    //
    // @Autowired
    // private MemberService memberService;

```

```

    @GetMapping("/addboard")
    public String requestAddBoardForm(@ModelAttribute("NewBoard") Board board) {
        return "board/addboard";
    }

    @PostMapping("/addboard")
    public String submitAddBoardForm(@ModelAttribute("NewBoard") Board board) {

        boardService.setNewBoard(board);

        // User user = userService.existUsername(board.getBwriter());
        //
        // String to = user.getUemail();
        // String subject = board.getBwriter() + " 님이 게시판에 글을 등록하셨습니다.";
        // String body = board.getBcontent();
        //
        // mailService.sendMail(to, subject, body);

        return "redirect:/board/list";
    }

    @GetMapping("/list")
    public String BoardList(Model model) {
        List<Board> list = boardService.getAllBoardList();
        model.addAttribute("boardList", list);

        return "board/list";
    }

    @GetMapping("/detail")
    public String requestBoardById(@RequestParam("bid") String bid, Model model) {
        // 주 게시물
        Board boardById = boardService.getBoardById(bid);
        model.addAttribute("board", boardById);

        // 답변
        List<Board> list = boardService.getReplyById(bid);
        int cnt = list.size();
        model.addAttribute("replyList", list);
        model.addAttribute("cnt", cnt);

        // 품을 띄우기 전에 조회수 하나 증가

        // Map<String, Object> map = new HashMap<String, Object>();
        //
        // map.put("check", "bview");
        // map.put("bid", bid);
        //
        // boardService.checkBoard(map);

        return "board/detail";
    }

    @ResponseBody
    @RequestMapping("/replaynew")
    public void replaynew(@RequestParam Map<String, Object> map) {

        String rpid = (String)map.get("rpwriter");
        String bpid = (String)map.get("bpwriter");

        boardService.replynewBoard(map);

        // User user = userService.existUsername(bpid);
        //
        // String to = user.getUemail();
        // String subject = bpid + " 님의 글에 댓글이 달렸습니다.";
        // String body = (String)map.get("bcontent");
        //
        // mailService.sendMail(to, subject, body);

    }
}

```

SQL

```

<?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="board">

```

```

<insert id="insert"
    parameterType="com.aglory.board.Board"
    useGeneratedKeys="true"
    keyProperty="bid">
<!CDATA[
    INSERT INTO gloryboard
    (btitle, bcontent, bcate, bstatus, bwriter)
    VALUES
    (#{$btitle}, #{$bcontent}, #{$bcate}, #{$bstatus}, #{$bwriter})
]]>
</insert>

<insert id="insert_reply"
    parameterType="hashMap"
    useGeneratedKeys="true"
    keyProperty="bid">
<!CDATA[
    INSERT INTO gloryboard
    (bpid, bcontent, bwriter)
    VALUES
    (#{$bid}, #{$bcontent}, #{$bwriter})
]]>
</insert>

<select id="select_list"
    resultType="com.aglory.board.Board">
<!CDATA[
    SELECT *
    FROM gloryboard
    WHERE bpid IS NULL
    ORDER BY bid
    DESC
]]>
</select>

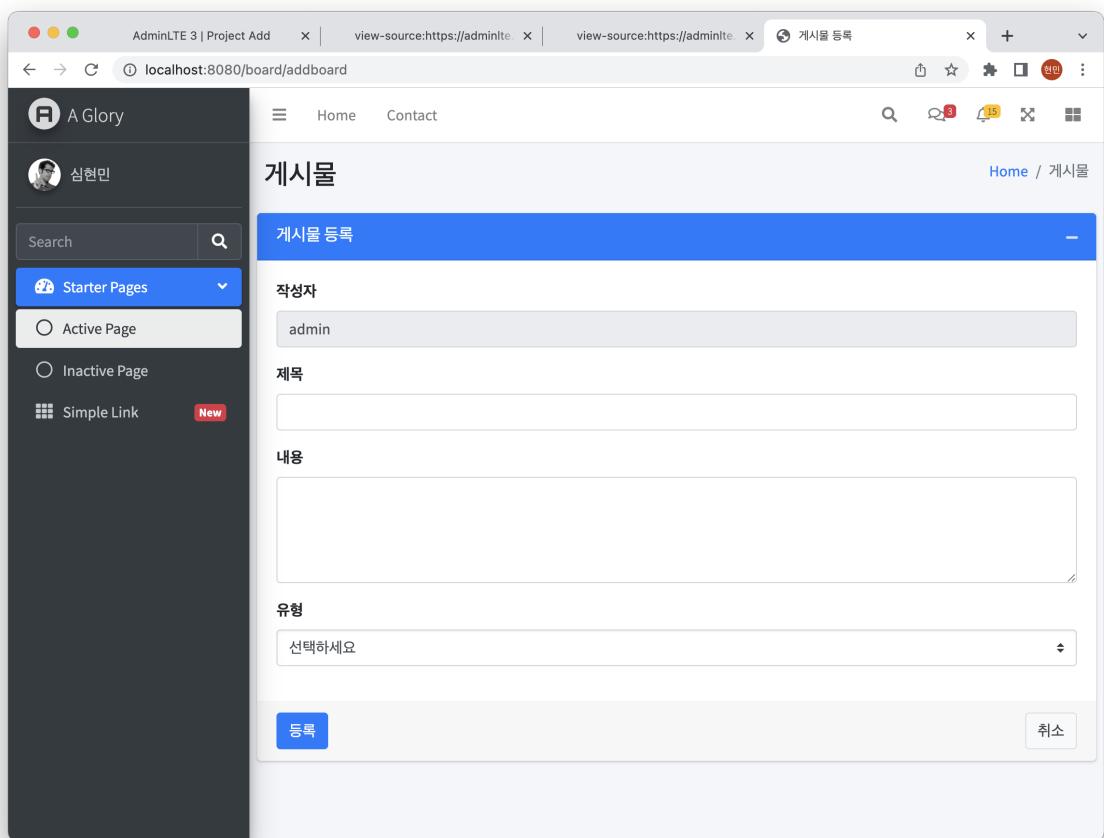
<select id="select_detail"
    parameterType="String"
    resultType="com.aglory.board.Board">
<!CDATA[
    select * from gloryboard where bid = #{$bid}
]]>
</select>

<select id="select_reply"
    parameterType="String"
    resultType="com.aglory.board.Board">
<!CDATA[
    SELECT *
    FROM gloryboard
    WHERE bpid = #{$bid}
    ORDER BY bid DESC
]]>
</select>

</mapper>

```

View



번호	작성자	제목	유형	작성일
1	admin	첫 게시물		2023-03-27 12:27:51
2	admin	2	기타	2023-03-27 12:29:26
3	admin	3	선택기능	2023-03-27 12:35:14

Notice

DTO

```

package com.aglory.notice;

public class Notice {

    private int nid;

    private String ntitle, ncontent, ncate, nwriter, ndate;

    public int getNid() {
        return nid;
    }

    public void setNid(int nid) {
        this.nid = nid;
    }

    public String getNtitle() {
        return ntitle;
    }

    public void setNtitle(String ntitle) {
        this.ntitle = ntitle;
    }

    public String getNcontent() {
        return ncontent;
    }

    public void setNcontent(String ncontent) {
        this.ncontent = ncontent;
    }
}

```

```

public String getNcate() {
    return ncate;
}

public void setNcate(String ncate) {
    this.ncate = ncate;
}

public String getNwriter() {
    return nwriter;
}

public void setNwriter(String nwriter) {
    this.nwriter = nwriter;
}

public String getNdate() {
    return ndate;
}

public void setNdate(String ndate) {
    this.ndate = ndate;
}

public Notice() {
}

public Notice(int nid, String ntitle, String ncontent, String ncate, String nwriter, String ndate) {
    this.nid = nid;
    this.ntitle = ntitle;
    this.ncontent = ncontent;
    this.ncate = ncate;
    this.nwriter = nwriter;
    this.ndate = ndate;
}
}

```

Repository

```

//C
void setNewNotice(Notice notice); //공지 작성

//R
List<Notice> getAllNoticeList(); //공지 전체 목록

Notice getNoticeById(String nid); //공지 하나
-----
@.Autowired
SqlSessionTemplate sqlSessionTemplate;

@Override
public void setNewNotice(Notice notice) {
    this.sqlSessionTemplate.insert("notice.insert", notice);
}

@Override
public List<Notice> getAllNoticeList() {
    return this.sqlSessionTemplate.selectList("notice.select_list");
}

@Override
public Notice getNoticeById(String nid) {
    return this.sqlSessionTemplate.selectOne("notice.select_detail", nid);
}

```

Service

```

void setNewNotice(Notice notice);

List<Notice> getAllNoticeList();

Notice getNoticeById(String nid);

```

```

-----
@.Autowired
NoticeRepository noticeRepository;

@Override
public void setNewNotice(Notice notice) {
    noticeRepository.setNewNotice(notice);
}

@Override
public List<Notice> getAllNoticeList() {
    return noticeRepository.getAllNoticeList();
}

@Override
public Notice getNoticeById(String nid) {
    return noticeRepository.getNoticeById(nid);
}

```

Controller

```

package com.aglory.notice;

import java.util.List;

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.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;

import com.aglory.board.Board;

@Controller
@RequestMapping("/notice")
public class NoticeController {

    @Autowired
    private NoticeService noticeService;

    @GetMapping("/addnotice")
    public String requestAddNoticeForm(@ModelAttribute("NewNotice") Notice notice) {
        return "notice/addnotice";
    }

    @PostMapping("/addnotice")
    public String submitAddNoticeForm(@ModelAttribute("NewNotice") Notice notice) {

        noticeService.setNewNotice(notice);

        return "redirect:/notice/list";
    }

    @GetMapping("/list")
    public String NoticeList(Model model) {
        List<Notice> list = noticeService.getAllNoticeList();
        model.addAttribute("noticeList", list);

        return "notice/list";
    }

    @GetMapping("/detail")
    public String requestNoticeById(@RequestParam("nid") String nid, Model model) {
        Notice noticeById = noticeService.getNoticeById(nid);
        model.addAttribute("notice", noticeById);

        return "notice/detail";
    }
}

```

SQL

```

<?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="notice">
    <insert id="insert"
        parameterType="com.aglory.notice.Notice"
        useGeneratedKeys="true"
        keyProperty="nid"><!-- 전부 -->
    <![CDATA[
        INSERT INTO glorynotice
        (ntitle, ncontent, ncate, nwritter)
        VALUES
        (#{{ntitle}}, #{{ncontent}}, #{{ncate}}, #{{nwritter}})
    ]]>
    </insert>

    <select id="select_list"
        resultType="com.aglory.notice.Notice">
    <![CDATA[
        SELECT *
        FROM glorynotice
        ORDER BY nid DESC
    ]]>
    </select>

    <select id="select_detail"
        parameterType="String"
        resultType="com.aglory.notice.Notice">
    <![CDATA[
        SELECT *
        FROM glorynotice
        WHERE nid = #{{nid}}
    ]]>
    </select>
</mapper>

```

View

The screenshot shows a web-based administrative interface for managing notices. The top navigation bar includes tabs for 'AdminLTE 3 | Project Add', 'view-source:https://adminlte...', 'view-source:https://adminlte...', and '공지 등록'. The main header features the title 'A Glory' and a user profile for '심현민'. A search bar and a sidebar menu are also present.

The main content area is titled '공지' (Notice) and displays a '공지 등록' (Post Notice) form. The form fields include:

- 작성자**: admin
- 제목**: (empty input field)
- 내용**: (empty input field)
- 유형**: 선택하세요 (dropdown menu)

At the bottom of the form are two buttons: '등록' (Register) and '취소' (Cancel). The footer contains copyright information: 'Copyright © A Glory. All rights reserved.' and 'Version 1.1.1'.

번호	제목	유형	작성일
1	게시물 등록 시 주의사항입니다.	기타	2023-03-27 14:36:49
2	게시물 등록 시 주의사항입니다.	기타	2023-03-27 14:37:24
3	1	버그처리	2023-03-27 14:37:47

Member

스프링 시큐리티에 맞게 생성 필요

DTO

```
package com.aglory.member;

public class Member {

    private int mid, enabled;

    private String username, password, authority, mname, memail, mtel, mdate;

    public int getMid() {
        return mid;
    }

    public void setMid(int mid) {
        this.mid = mid;
    }

    public int getEnabled() {
        return enabled;
    }

    public void setEnabled(int enabled) {
        this.enabled = enabled;
    }

    public String getUsername() {
        return username;
    }
}
```

```

public void setUsername(String username) {
    this.username = username;
}

public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}

public String getAuthority() {
    return authority;
}

public void setAuthority(String authority) {
    this.authority = authority;
}

public String getMname() {
    return mname;
}

public void setMname(String mname) {
    this.mname = mname;
}

public String getMemail() {
    return memail;
}

public void setMemail(String memail) {
    this.memail = memail;
}

public String getMtel() {
    return mtel;
}

public void setMtel(String mtel) {
    this.mtel = mtel;
}

public String getMdate() {
    return mdate;
}

public void setMdate(String mdate) {
    this.mdate = mdate;
}

public Member() {

}

public Member(int mid, int enabled, String username, String password, String authority, String mname, String memail,
    String mtel, String mdate) {
    super();
    this.mid = mid;
    this.enabled = enabled;
    this.username = username;
    this.password = password;
    this.authority = authority;
    this.mname = mname;
    this.memail = memail;
    this.mtel = mtel;
    this.mdate = mdate;
}
}

```

Repository

```

//C
void setNewMember(Member member); //회원 가입

//R

```

```

List<Member> getAllMemberList(); //회원 전체 목록

Member getMemberById(String mid); //회원 하나
-----
@Autowired
SqlSessionTemplate sqlSessionTemplate;

@Override
public void setNewMember(Member member) {
    this.sqlSessionTemplate.insert("member.insert", member);
}

@Override
public List<Member> getAllMemberList() {
    return this.sqlSessionTemplate.selectList("member.select_list");
}

@Override
public Member getMemberById(String mid) {
    return this.sqlSessionTemplate.selectOne("member.select_detail", mid);
}

```

Service

```

void setNewMember(Member member);

List<Member> getAllMemberList();

Member getMemberById(String mid);
-----
@Autowired
MemberRepository memberRepository;

@Override
public void setNewMember(Member member) {
    memberRepository.setNewMember(member);
}

@Override
public List<Member> getAllMemberList() {
    return memberRepository.getAllMemberList();
}

@Override
public Member getMemberById(String mid) {
    return memberRepository.getMemberById(mid);
}

```

Controller

```

package com.aglory.member;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;

@Controller
@RequestMapping("/member")
public class MemberController {

    @Autowired
    MemberService memberService;

    // @Autowired
    // BCryptPasswordEncoder bcryptPasswordEncoder;

    @GetMapping("/addmember")
}

```

```

public String requestAddMemberForm(@ModelAttribute("NewMember") Member member) {
    return "member/addmember";
}

@PostMapping("/addmember")
public String submitAddMemberForm(@ModelAttribute("NewMember") Member member) {
//    //스프링은 반드시 password를 암호화 하여 저장해야만 로그인을 할수 있는게 기본이다.
//    String encodedPassword = bryptPasswordEncoder.encode(member.getPassword());
//    member.setPassword(encodedPassword);

    memberService.setNewMember(member);

    return "redirect:/member/list";
}

@GetMapping("/list")
public String memberList(Model model) {
    List<Member> list = memberService.getAllMemberList();
    model.addAttribute("memberList", list);

    return "member/list";
}

@GetMapping("/detail")
public String requestMemberById(@RequestParam("mid") String mid, Model model) {
    Member memberById = memberService.getMemberById(mid);
    model.addAttribute("member", memberById);

    return "member/detail";
}
}

```

SQL

```

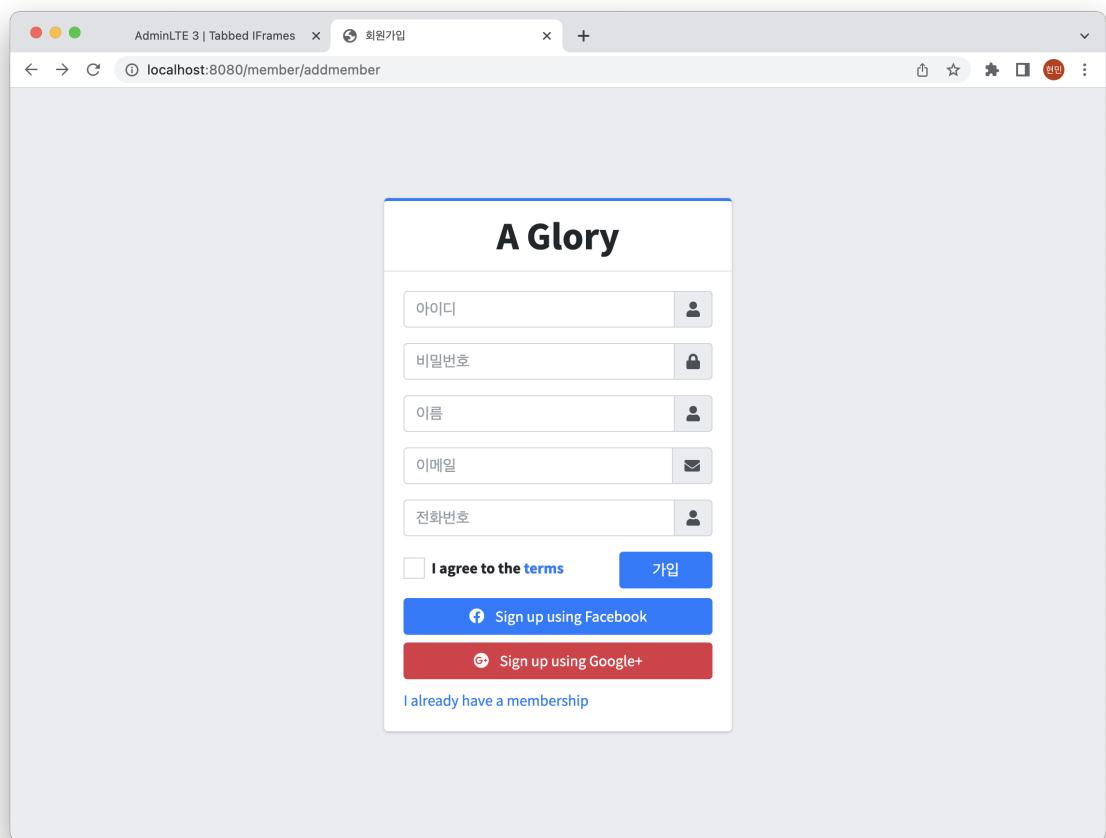
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="member">
    <insert id="insert"
        parameterType="com.aglory.member.Member"
        useGeneratedKeys="true"
        keyProperty="username"><!-- 전부 -->
    <![CDATA[
        INSERT INTO glorymember
        (username, password, mname, memail, mtel)
        VALUES
        (#{$username}, #{$password}, #{$mname}, #{$memail}, #{$mtel})
    ]]>
    </insert>

    <select id="select_list"
        resultType="com.aglory.member.Member">
    <![CDATA[
        SELECT *
        FROM glorymember
        ORDER BY mid DESC
    ]]>
    </select>

    <select id="select_detail"
        parameterType="String"
        resultType="com.aglory.member.Member">
    <![CDATA[
        SELECT *
        FROM glorymember
        WHERE mid = #{$mid}
    ]]>
    </select>
</mapper>

```

View



The screenshot shows a web application interface for managing users. On the left, there is a sidebar with a dark theme containing:

- A logo for "A Glory".
- A user profile picture and name "심현민".
- A search bar with placeholder "Search" and a magnifying glass icon.
- A dropdown menu:
 - Starter Pages** (selected)
 - Active Page
 - Inactive Page
- A "Simple Link" button with a "New" badge.

The main content area has a light background and displays the following information:

회원 (Members) / [Home](#) / [회원](#)

회원 목록 (Member List)

Buttons: Copy, CSV, Excel, PDF, Print, Column visibility ▾, Search: [input field].

Table:

번호	아이디	이름	Auth	Enabled
1	hyunmin	현민	ROLE_USER	1
	번호	아이디	이름	Auth

Showing 1 to 1 of 1 entries

Buttons: Previous, **1**, Next.