PORTFOLIO

팀프로젝트 B조(권용규,권도균,류경문,이승훈,이강호,주지환)

◆ 2022.02.27~
2022.02.27

PROJECT X
CONTENTS

프로젝트 개요

스프링으로 나만의 페이지를 만들어 페이지를 이용해 이제껏 배웠던 프로젝트들을 올리고 관리도하면서 이것저것 적용해보고 싶었습니다. 페이지는 어떻게 불러와 지는지, 프론트와 백에서는 어떻게 데이터를 주고받는지 공부하며 회원가입과 로그인, 게시판페이지를 만들어 보았습니다.

spring 주요기능

MY BATIS(XML)

my bartis를 활용해 xml만으로 sql문을 작성해 데이터를 연결해주 었습니다. mybaris를 사용하면서 어노테이션과 인터페이스만으로 sql문을 작성할 수 있어서 중복되는 코드양을 확 줄일 수 있었습니다

데이터 객체(VO)

VO클레스를 만들어 로그인과 게시판을 만들때 들어갈 데이터의 변수명과 맞추어 데이터를 주고 받을 수 있도록 만들었습니다.

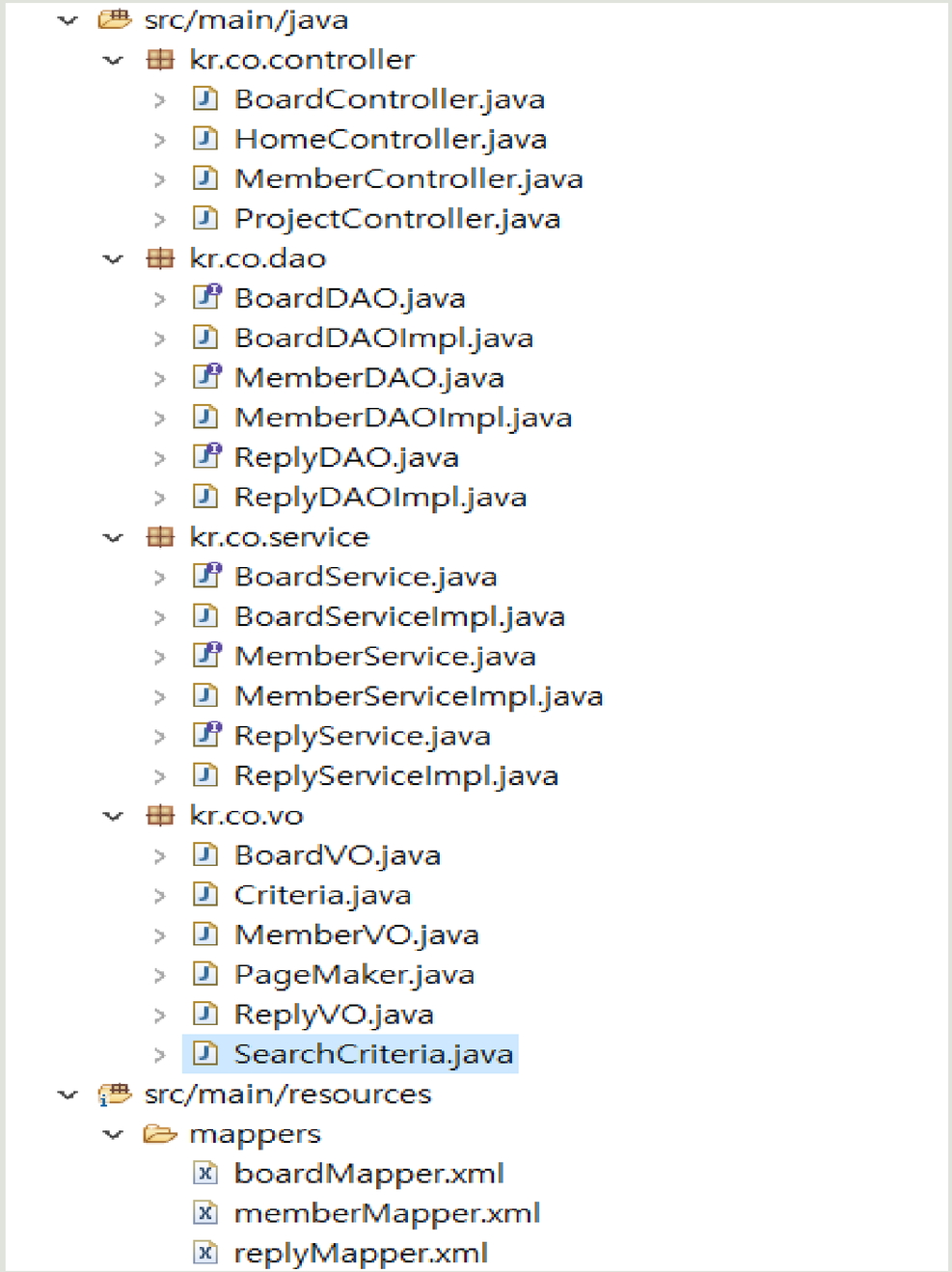
DAO,IMPL(implememt)

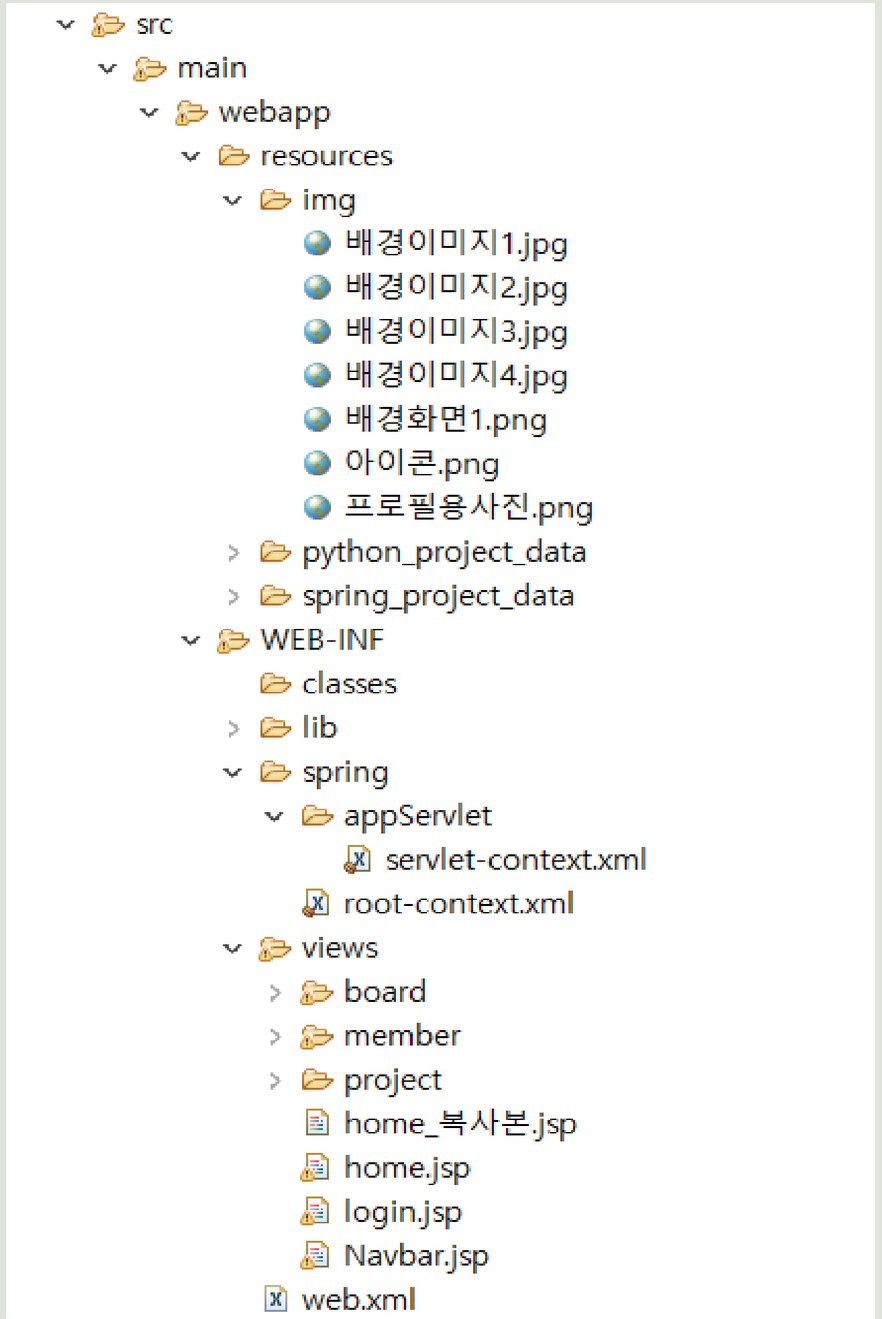
DAO와 implement 클래스를 따로 만들어 사용자가 원하는 데이터를 부를때 VO객체를 사용하여 데이터를 서버로부터 불러올 수있게 만들었습니다.

CONTROLLER & SERVICE

view단에서 경로가 입력되면 해당 컨트롤러에 맵핑되는 메소드가 실행되면서 미리 만들어둔 서비스 클래스에 사용자의 요구사항을 처리할수 있도록 만들었습니다.

코드의 분리


src/main/java
kr.co.controller
BoardController.java
HomeController.java
MemberController.java
ProjectController.java
kr.co.dao
BoardDAO.java
BoardDAOImpl.java
MemberDAO.java
MemberDAOImpl.java
ReplyDAO.java
ReplyDAOImpl.java
kr.co.service
BoardService.java
BoardServiceImpl.java
MemberService.java
MemberServiceImpl.java
ReplyService.java
ReplyServiceImpl.java
kr.co.vo
BoardVO.java
Criteria.java
MemberVO.java
PageMaker.java
ReplyVO.java
SearchCriteria.java
src/main/resources
mappers
boardMapper.xml
memberMapper.xml
replyMapper.xml


src
main
webapp
resources
img
배경이미지1.jpg
배경이미지2.jpg
배경이미지3.jpg
배경이미지4.jpg
배경화면1.png
아이콘.png
프로필용사진.png
python_project_data
spring_project_data
WEB-INF
classes
lib
spring
appServlet
servlet-context.xml
root-context.xml
views
board
member
project
home_복사본.jsp
home.jsp
login.jsp
Navbar.jsp
web.xml

```
    <!-- 오라클 접속 -->
    <bean class="org.springframework.jdbc.datasource.DriverManagerDataSource" id="dataSource">
        <property name="driverClassName" value="net.sf.log4jdbc.sql.jdbcapi.DriverSpy"/>
        <property name="url" value="jdbc:log4jdbc:oracle:thin:@localhost:1521/xe"/>
        <property name="username" value="system"/>
        <property name="password" value="1234"/>
    </bean>

    <!-- Mybatis 연동 -->
    <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
         <property name="dataSource" ref="dataSource"></property>

          <!-- mybatis-config.xml이 스프링 시작될때 같이 실행되도록 설정하기  -->
         <property name="configLocation" value="classpath:/mybatis-config.xml"></property>

         <!-- XML mapper를 인식하게 설정 -->
         <property name="mapperLocations" value="classpath:mappers/**/*Mapper.xml"/>
    </bean>

    <!-- SQLSessionTemplate설정하기 DAO인터페이스를 만들었기때문에 Mybatis에서 DAO인터페이스를 구현하기위해서 필요한작업 -->
    <bean id="sqlSession" class="org.mybatis.spring.SqlSessionTemplate" destroy-method="clearCache">
        <constructor-arg name="sqlSessionFactory" ref="sqlSessionFactory"></constructor-arg>
    </bean>

     <!-- scan -->
    <context:component-scan base-package="kr.co.service"></context:component-scan>
    <context:component-scan base-package="kr.co.dao"></context:component-scan>
    <context:component-scan base-package="kr.co.vo"></context:component-scan>
</beans>
```

MYbartis 연동과 데이터베이스와 연결하기 위한 root-servlet 설정 코드

```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="boardMapper">
    <!-- 게시판 글 작성 -->
    <insert id="insert">
        INSERT INTO MP_BOARD(    BNO
                            , TITLE
                            , CONTENT
                            , WRITER  )
                    VALUES(    MP_BOARD_SEQ.NEXTVAL
                            , #{title}
                            , #{content}
                            , #{writer}  )
    </insert>

    <!-- 게시판 글 목록 -->
    <select id="listPage" resultType="kr.co.vo.BoardVO" parameterType="kr.co.vo.SearchCriteria">
        SELECT  BNO,
                TITLE,
                CONTENT,
                WRITER,
                REGDATE
         FROM (
                SELECT BNO,
                       TITLE,
                       CONTENT,
                       WRITER,
                       REGDATE,
                       ROW_NUMBER() OVER(ORDER BY BNO DESC) AS RNUM
                FROM MP_BOARD
                WHERE 1=1
                   <include refid="search"></include>
                            ) MP
        WHERE RNUM BETWEEN #{rowStart} AND #{rowEnd}
        ORDER BY BNO DESC
    </select>
```

MYbartis를 사용한 mapper 파일의 SQL 작성 코드<1>

→코드가 다 나오진 않지만 select와 insert, delete, update 코드를 넣어 crud기능을 사용합니다.

MYbartis를 사용한 mapper 파일의 SQL 작성 코드< 2 >

```
<!-- 게시판 제목으로 글 접속 -->
<select id="read" parameterType="int" resultType="kr.co.vo.BoardVO">
    SELECT  BNO
            , TITLE
            , CONTENT
            , WRITER
            , REGDATE
        FROM MP_BOARD
        WHERE BNO = #{bno}
</select>

<!-- 게시판 글 수정 -->
<update id="update" parameterType="kr.co.vo.BoardVO">
    UPDATE MP_BOARD
       SET TITLE    =   #{title},
           CONTENT  =   #{content}
     WHERE BNO = #{bno}
</update>

<!-- 게시판 글 삭제 -->
<delete id="delete" parameterType="int">
    DELETE
      FROM MP_BOARD
     WHERE BNO = #{bno}
</delete>

<!-- 게시판 총 갯수 -->
<select id="listCount" parameterType="kr.co.vo.SearchCriteria" resultType="int">
    SELECT COUNT(BNO)
       FROM MP_BOARD
       WHERE 1=1
    <include refid="search"></include>
       AND BNO > 0
</select>

  <!-- 검색 목록 기능 -->
  <sql id="search">
      <if test="searchType != null">
          <if test="searchType == 't'.toString()">AND TITLE LIKE '%' || #{keyword} || '%'</if>
          <if test="searchType == 'c'.toString()">AND CONTENT LIKE '%' || #{keyword} || '%'</if>
          <if test="searchType == 'w'.toString()">AND WRITER LIKE '%' || #{keyword} || '%'</if>
          <if test="searchType == 'tc'.toString()">AND (TITLE LIKE '%' || #{keyword} || '%') or (CONTENT LIKE '%' || #{keyword} || '%')</if>
      </if>
  </sql>
</mapper>
```

※mapper에서 sql문의 id는 DAO에서 사용할 메서드와 이름을 맞춰주어 헷깔림을 방지해줍니다

```java
public class BoardVO {
    private int bno;
    private String title;
    private String content;
    private String writer;
    private Date regdate;

    public int getBno() {
        return bno;
    }
    public void setBno(int bno) {
        this.bno = bno;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getContent() {
        return content;
    }
    public void setContent(String content) {
        this.content = content;
    }
    public String getWriter() {
        return writer;
    }
    public void setWriter(String writer) {
        this.writer = writer;
    }
    public Date getRegdate() {
        return regdate;
    }
    public void setRegdate(Date regdate) {
        this.regdate = regdate;
    }

}
```

게시판의 VO부분 입니다.

```java
public interface BoardDAO {

    // 게시글 작성
    public void write(BoardVO boardVO) throws Exception;

    // 게시물 목록 조회
    public List<BoardVO> list(SearchCriteria scri) throws Exception;

    // 게시물 총 갯수
    public int listCount(SearchCriteria scri) throws Exception;

    // 게시물 조회
    public BoardVO read(int bno) throws Exception;

    // 게시물 수정
    public void update(BoardVO boardVO) throws Exception;

    // 게시물 삭제
    public void delete(int bno) throws Exception;
}
```

※mapper의 id와 이름이 같은 메서드 모습

```java
@Repository
public class BoardDAOImpl implements BoardDAO {

    @Inject
    private SqlSession sqlSession;

    // 게시글 작성
    @Override
    public void write(BoardVO boardVO) throws Exception {
        sqlSession.insert("boardMapper.insert", boardVO);
    }

    // 게시물 목록 조회
    @Override
    public List<BoardVO> list(SearchCriteria scri) throws Exception {

        return sqlSession.selectList("boardMapper.listPage",scri);
    }

    // 게시물 총 갯수
    @Override
    public int listCount(SearchCriteria scri) throws Exception{

        return sqlSession.selectOne("boardMapper.listCount",scri);
    }
```

게시판의 DAO부분과 DAOImpl부분<1>

```java
    // 게시물 조회
    @Override
    public BoardVO read(int bno) throws Exception {

        return sqlSession.selectOne("boardMapper.read", bno);
    }

    // 게시물 수정
    @Override
    public void update(BoardVO boardVO) throws Exception {

        sqlSession.update("boardMapper.update", boardVO);
    }

    // 게시물 삭제
    @Override
    public void delete(int bno) throws Exception {

        sqlSession.delete("boardMapper.delete", bno);
    }
}
```

게시판의 DAO부분과 DAOImpl부분< 2 >

```java
public interface BoardService {

    // 게시글 작성
    public void write(BoardVO boardVO) throws Exception;

    // 게시물 목록 조회
    public List<BoardVO> list(SearchCriteria scri) throws Exception;

    // 게시물 총 갯수
    public int listCount(SearchCriteria scri) throws Exception;

    // 게시물 조회
    public BoardVO read(int bno) throws Exception;

    // 게시물 수정
    public void update(BoardVO boardVO) throws Exception;

    // 게시물 삭제
    public void delete(int bno) throws Exception;
}
```

게시판의 Service부분과 ServiceImpl부분 <1>

```java
@Service
public class BoardServiceImpl implements BoardService {

    @Inject
    private BoardDAO dao;

    // 게시글 작성
    @Override
    public void write(BoardVO boardVO) throws Exception {
        dao.write(boardVO);
    }

    // 게시물 목록 조회
    @Override
    public List<BoardVO> list(SearchCriteria scri) throws Exception {

        return dao.list(scri);
    }
    // 게시물 총 갯수
    @Override
    public int listCount(SearchCriteria scri) throws Exception{

        return dao.listCount(scri);
    }

    // 게시물 조회
    @Override
    public BoardVO read(int bno) throws Exception {

        return dao.read(bno);
    }

    // 게시물 목록 수정
    @Override
    public void update(BoardVO boardVO) throws Exception {

        dao.update(boardVO);
    }

    // 게시물 목록 삭제
    @Override
    public void delete(int bno) throws Exception {

        dao.delete(bno);
    }
}
```

게시판의 Service부분과 ServiceImpl부분 < 2 >

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

    private static final Logger logger = LoggerFactory.getLogger(BoardController.class);

    @Inject
    BoardService service;

    @Inject
    ReplyService replyService;

    // 게시판 글 작성 화면
    @RequestMapping(value = "/board/writeView", method = RequestMethod.GET)
    public void writeView() throws Exception{
        logger.info("writeView");

    }

    // 게시판 글 작성
    @RequestMapping(value = "/board/write", method = RequestMethod.POST)
    public String write(BoardVO boardVO) throws Exception{
        logger.info("write");

        service.write(boardVO);

        return "redirect:/board/list";
    }
```

게시판의 Controller 부분 <1>

```java
// 게시판 목록 조회
@RequestMapping(value = "/list", method = RequestMethod.GET)
public String list(Model model, @ModelAttribute("scri") SearchCriteria scri) throws Exception{
    Logger.info("list");

    model.addAttribute("list", service.list(scri));

    PageMaker pageMaker = new PageMaker();
    pageMaker.setCri(scri);
    pageMaker.setTotalCount(service.listCount(scri));

    model.addAttribute("pageMaker", pageMaker);

    return "board/list";

}

// 게시판 조회
@RequestMapping(value = "/readView", method = RequestMethod.GET)
public String read(BoardVO boardVO, @ModelAttribute("scri") SearchCriteria scri, Model model) throws Exception{
    Logger.info("read");

    model.addAttribute("read", service.read(boardVO.getBno()));
    model.addAttribute("scri", scri);

    List<ReplyVO> replyList = replyService.readReply(boardVO.getBno());
    model.addAttribute("replyList", replyList);

    return "board/readView";
}
```

게시판의 Controller 부분 < 2 >

```java
// 게시판 수정뷰
@RequestMapping(value = "/updateView", method = RequestMethod.GET)
public String updateView(BoardVO boardVO, @ModelAttribute("scri") SearchCriteria scri, Model model) throws Exception{
    logger.info("updateView");

    model.addAttribute("update", service.read(boardVO.getBno()));
    model.addAttribute("scri", scri);

    return "board/updateView";
}

// 게시판 수정
@RequestMapping(value = "/update", method = RequestMethod.POST)
public String update(BoardVO boardVO, @ModelAttribute("scri") SearchCriteria scri, RedirectAttributes rttr) throws Exception{
    logger.info("update");

    service.update(boardVO);

    rttr.addAttribute("page",scri.getPage());
    rttr.addAttribute("perPageNum", scri.getPerPageNum());
    rttr.addAttribute("searchType", scri.getSearchType());
    rttr.addAttribute("keyword", scri.getKeyword());

    return "redirect:/board/list";
}
```

게시판의 Controller 부분 < 3 >

```java
// 게시판 삭제
@RequestMapping(value = "/delete", method = RequestMethod.POST)
public String delete(BoardVO boardVO, @ModelAttribute("scri") SearchCriteria scri, RedirectAttributes rttr) throws Exception{
    logger.info("delete");

    service.delete(boardVO.getBno());

    rttr.addAttribute("page",scri.getPage());
    rttr.addAttribute("perPageNum", scri.getPerPageNum());
    rttr.addAttribute("searchType", scri.getSearchType());
    rttr.addAttribute("keyword", scri.getKeyword());

    return "redirect:/board/list";
}
```

게시판의 Controller 부분 < 4 >

Q & A

Thanks for

Your Watching.