Spring Web MVC

MVC : Model, View, Controller의 약자입니다.

현재 가장 많이 사용하는 개발 패턴으로 서비스를 위한 각 부분을 분리해 만들고 이를 통해 개발 및 유지 보수의 효율성을 높이는 것이 목적입니다.

- Model : 데이터를 관리하는 부분
- View : 눈에 보이는 부분을 구성하는 부분
- Controller : 요청에 따른 코드 흐름을 제어하는 부분

(1)	SpringConfigClass {
	환경설정클래스 -ServletAppContext
	환경Root 빈 정의 -RootAppContext
	}
		: AbstractAnnotationConfigDispatcherServletInitializer 상속

(2)	@Configuration ServletAppContext { @ComponentScan("kr.co.korea.controller") @ComponentScan("kr.co.korea.dao") @ComponentScan("kr.co.korea.service") @PropertySource("/WEB-INF/properties/db.properties") @Bean -db접속,쿼리접속,쿼리실행	: WebMvcConfigurer 인터페이스 상속
(3)	@Configuration RootAppContext { @Bean("loginUserBean") @SessionScope	

(1) @Bean : 메서드를 통해 반환하는 객체를 Bean으로 등록
ex) [kr.co.korea.config]패키지의 ServletAppContext 클래스의 DB접속정보, 쿼리문, mapper

(2) @Component : 개발자가 만든 클래스의 객체를 생성하여 Bean으로 등록
ex) [kr.co.korea.beans]패키지의 DataBean 클래스의 객체를 생성하여 등록할때

(3) @Controller : Component의 일종으로 사용자 요청에 따라 **자동으로 호출되는 메서드**를 가지고 있는 Bean을 등록. jsp로 리턴 ex) [kr.co.korea.controller]패키지의 HomeController클래스, TestController클래스

(4) @RestController : Component의 일종으로 사용자 요청에 따라 **자동으로 호출되는 메서드**를 가지고 있는 Bean을 등록. Restful API 서버 구성 시 사용. 전달 글자 그대로 리턴

(5) @ControllerAdvice : 예외가 발생했을 때 사용할 Global Exception Handler로 사용할 Bean을 등록
ex) [kr.co.korea.exception]패키지의 GlobalException 클래스 만들때...

(6) @Repository : dao에서 @Repository로 클래스를 빈 정의 해두고 service의 @Service로 정의한 Bean에서 @Autowired 로 주입받아 사용. 이 Bean은 데이터베이스와 관련된 작업을 구현. @Component로 정의한 Bean과 차이가 없음.
ex) [kr.co.korea.dao]패키지 TopMenuDao클래스 만들때

(7) @Service : service에서 빈 정의 해두고 Controller에서 @Autowired 이용하여 주입받아 사용. @Component로 정의한 Bean과 차이는 없음. ex) [kr.co.korea.service]패키지의 UserService클래스, TopMenuService클래스 만들때...

(8) @Configuration : 설정클래스임을 지정
ex) [kr.co.korea.config]패키지의 ServletAppContext.java에서 **설정 클래스**임을 지정

(9) @EnableWebMvc : 설정클래스에서 Spring MVC환경을 구성할 수 있음
ex) [kr.co.korea.config]패키지의 ServletAppContext.java

(10) @ComponentScan("패키지") : 스캔할 패키지 지정
ex) [kr.co.korea.config]패키지의 ServletAppContext.java 에서

(11) @PropertySource(), @PropertySources() : properties파일을 주입받기
ex) [kr.co.korea.config]패키지의 ServletAppContext.java

(12) @Value : properties파일의 값을 주입받기
ex) [kr.co.korea.config]패키지의 ServletAppContext.java

1. Java방법으로 스프링 설정

(1) project.sql

```
create sequence user_seq
start with 0
increment by 1
minvalue 0;

create sequence content_seq
start with 0
increment by 1
minvalue 0;

create table board_info_table(
	board_info_idx number constraint BOARD_INFO_PK primary key,
	board_info_name varchar2(500) not null
);


create table user_table(
	user_idx number constraint USER_PK primary key,
	user_name varchar2(50) not null,
	user_id varchar2(100) not null,
	user_pw varchar2(100) not null
);

create table content_table(
	content_idx number constraint CONTENT_PK primary key,
	content_subject varchar2(500) not null,
	content_text long not null,
	content_file varchar2(500),
	content_writer_idx number not null
	                    constraint CONTENT_FK1 references user_table(user_idx),
	content_board_idx number not null
	                   constraint CONTENT_FK2 references board_info_table(board_info_idx),
	content_date date not null
);
```

```
insert into board_info_table(board_info_idx, board_info_name) values (1, '1자유게시판');
insert into board_info_table(board_info_idx, board_info_name) values (2, '2게시판');
insert into board_info_table(board_info_idx, board_info_name) values (3, '3게시판');
insert into board_info_table(board_info_idx, board_info_name) values (4, '4게시판');

commit;
```

게시판 정보 테이블(board_info_table)

필드명(논리)	필드명	타입	NULL	PK	FK	UQ
게시판 번호	board_info_idx	number	X	O	X	O
게시판 이름	board_info_name	varchar2(500)	X	X	X	X

사용자 정보 테이블(user_table)

필드명(논리)	필드명	타입	NULL	PK	FK	UQ
사용자 번호	user_idx	number	X	O	X	O
사용자 이름	user_name	varchar2(50)	X	X	X	X
사용자 아이디	user_id	varchar2(100)	X	X	X	O
사용자 비밀번호	user_pw	varchar2(100)	X	X	X	X

게시글 테이블(content_table)

필드명(논리)	필드명	타입	NULL	PK	FK	UQ
게시글 인덱스	content_idx	number	X	O	X	O
게시글 제목	content_subject	varchar2(500)	X	X	X	X
게시글 내용	content_text	long	X	X	X	X
첨부파일	content_file	varchar2(500)	O	X	X	X
게시글 작성자 인덱스	content_writer_idx	number	X	X	user_table(user_idx)	X
게시판 인덱스	content_board_idx	number	X	X	board_info_table(board_info_idx)	X
작성날짜	content_date	date	X	X	X	X

(2) pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>
        <groupId>webProjectJava</groupId>
        <artifactId>webProjectJava</artifactId>
        <version>0.0.1-SNAPSHOT</version>
        <packaging>war</packaging>
        <build>
                <sourceDirectory>src</sourceDirectory>
                <plugins>
                        <plugin>
                                <artifactId>maven-compiler-plugin</artifactId>
                                <version>3.8.1</version>
                                <configuration>
                                        <source>1.8</source>
                                        <target>1.8</target>
                                </configuration>
                        </plugin>
                        <plugin>
                                <artifactId>maven-war-plugin</artifactId>
                                <version>3.2.3</version>
                <configuration>
                    <warSourceDirectory>WebContent</warSourceDirectory>
                </configuration>
            </plugin>
        </plugins>
    </build>
    <!-- 라이브러리 버전관리 -->
    <properties>
        <javax.servlet-version>4.0.1</javax.servlet-version>
        <javax.servlet.jsp-version>2.3.3</javax.servlet.jsp-version>
        <javax.servlet-jstl-version>1.2</javax.servlet-jstl-version>
        <org.springframework-version>5.2.8.RELEASE</org.springframework-version>
    </properties>

    <!-- 라이브러리 셋팅 -->
    <dependencies>
        <!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>javax.servlet-api</artifactId>
            <version>${javax.servlet-version}</version>
            <scope>provided</scope>
        </dependency>
        <!-- https://mvnrepository.com/artifact/javax.servlet.jsp/javax.servlet.jsp-api -->
        <dependency>
            <groupId>javax.servlet.jsp</groupId>
            <artifactId>javax.servlet.jsp-api</artifactId>
            <version>${javax.servlet.jsp-version}</version>
            <scope>provided</scope>
        </dependency>
        <!-- https://mvnrepository.com/artifact/javax.servlet/jstl -->
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>jstl</artifactId>
            <version>${javax.servlet-jstl-version}</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-webmvc</artifactId>
            <version>${org.springframework-version}</version>
        </dependency>

        <dependency>
            <groupId>javax.validation</groupId>
            <artifactId>validation-api</artifactId>
            <version>2.0.1.Final</version>
        </dependency>

        <dependency>
            <groupId>org.hibernate.validator</groupId>
            <artifactId>hibernate-validator</artifactId>
            <version>6.1.2.Final</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-jdbc</artifactId>
            <version>5.2.8.RELEASE</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/org.apache.commons/commons-dbcp2 -->
        <dependency>
            <groupId>org.apache.commons</groupId>
            <artifactId>commons-dbcp2</artifactId>
            <version>2.7.0</version>
        </dependency>

        <!-- https://mvnrepository.com/artifact/org.mybatis/mybatis-spring -->
        <dependency>
            <groupId>org.mybatis</groupId>
            <artifactId>mybatis-spring</artifactId>
            <version>2.0.5</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/org.mybatis/mybatis -->
        <dependency>
```

```
            <groupId>org.mybatis</groupId>
            <artifactId>mybatis</artifactId>
            <version>3.5.5</version>
        </dependency>

    </dependencies>

</project>
```

1. mybatis 개념

(1) mapper작성:쿼리문

[kr.co.korea.mapper] UserMapper.java 쿼리문 작성

쿼리문작성
(java방식)
	public interface UserMapper {
	@Select("select user_idx, user_name " +
	"from user_table " +
	"where user_id=#{user_id} and user_pw=#{user_pw}")
	UserBean getLoginUserInfo(UserBean tempLoginUserBean);
	}

쿼리문작성 (xml방식)	<mapper namespace="user"> <select id="getLoginUserInfo" parameterType="kr.co.korea.beans.UserBean" resultType="kr.co.korea.beans.UserBean"> <![CDATA[select user_idx, user_name from user_table where user_id=#{user_id} and user_pw=#{user_pw}]]> </select> </mapper>

(2) dao작성:mapper와 직접적으로 연결해주는 메소드정의

[kr.co.korea.dao] UserDao.java

Dao
(java방식)
	@Repository
	public class UserDao {
	@Autowired
	private UserMapper userMapper;
	
	public UserBean getLoginUserInfo(UserBean tempLoginUserBean) {
	return userMapper.getLoginUserInfo(tempLoginUserBean);
	}

Dao
(xml방식)
	@Repository
	public class UserDao {
	@Autowired
	private SqlSessionTemplate sqlSessionTemplate;
	
	public UserBean getLoginUserInfo(UserBean tempLoginUserBean) {
	return sqlSessionTemplate.selectOne("user.getLoginUserInfo", tempLoginUserBean);
	}

(3) service작성:dao의 메소드 가져와서 새로운 변형을 위한 메소드정의

[kr.co.korea.service] UserService.java

Service	@Service
	public class UserService {
	
	@Autowired
	private UserDao userDao;
	
	//UserDao의 getLoginUserInfo() 호출하도록 정의
	public void getLoginUserInfo(UserBean tempLoginUserBean) {
	
	UserBean tempLoginUserBean2 = userDao.getLoginUserInfo(tempLoginUserBean);
	
	if(tempLoginUserBean2 != null) { //db에서 select된게 있으면
	loginUserBean.setUser_idx(tempLoginUserBean2.getUser_idx());
	loginUserBean.setUser_name(tempLoginUserBean2.getUser_name());
	loginUserBean.setUserLogin(true);
	}
	}
	}

(4) controller작성:service의 메소드 가져와서 jsp파일(views)로 이동하는 메소드 정의

[kr.co.korea.controller] UserController.java

<table>
<tr>
<td>Controller</td>
<td><pre>
@Controller
@RequestMapping("/user")
public class UserController {

        //userService 객체로 주입
        @Autowired
        private UserService userService;

        @PostMapping("/login_pro")
        public String login_pro(@Valid @ModelAttribute("tempLoginUserBean") UserBean
tempLoginUserBean, BindingResult result) {

                if(result.hasErrors()) {
                        return "user/login";
                }

                userService.getLoginUserInfo(tempLoginUserBean);

                if(loginUserBean.isUserLogin() == true) {
                        return "user/login_success";
                } else {
                        return "user/login_fail";
                }
        }
}
</pre></td>
</tr>
</table>

(1) java 방식 설정

1) db.properties

[WebContent-WEB-INF-properties폴더]

```
db.classname=oracle.jdbc.OracleDriver
db.url=jdbc:oracle:thin:@localhost:1521:xe
db.username=system
db.password=123456
```

2) BoardMapper.java 인터페이스

[kr.co.korea.mapper]

```
package kr.co.korea.mapper;

public interface BoardMapper {

}
```

3) ServletAppContext.java

```
//1)
@PropertySource("/WEB-INF/properties/db.properties")
public class ServletAppContext implements WebMvcConfigurer {

//2)
        @Value("${db.classname}")
        private String db_classname;

        @Value("${db.url}")
        private String db_url;

        @Value("${db.username}")
        private String db_username;

        @Value("${db.password}")
        private String db_password;


//3) 데이터베이스 접속 정보 관리 (org.apache.commons.dbcp2.BasicDataSource)
        @Bean
        public BasicDataSource dataSource() {
                BasicDataSource source = new BasicDataSource();
                source.setDriverClassName(db_classname);
                source.setUrl(db_url);
                source.setUsername(db_username);
                source.setPassword(db_password);
                // System.out.println("db connect");
                return source;
        }

//4) 쿼리문과 접속 관리하는 객체
        @Bean
        public SqlSessionFactory factory(BasicDataSource source) throws Exception {
                SqlSessionFactoryBean factoryBean = new SqlSessionFactoryBean();
                factoryBean.setDataSource(source);
                SqlSessionFactory factory = factoryBean.getObject();
                // System.out.println("sql");
                return factory;
        }

//5) 쿼리문 실행을 위한 객체 (BoardMapper생성할것)
        @Bean
        public MapperFactoryBean<BoardMapper> getBoardMapper(SqlSessionFactory factory) throws
Exception {
                MapperFactoryBean<BoardMapper> factoryBean = new
MapperFactoryBean<BoardMapper>(BoardMapper.class);
                factoryBean.setSqlSessionFactory(factory);
                return factoryBean;
        }
```

(2) xml 방식 설정

1) db.properties

[WebContent-WEB-INF-properties폴더]

```
db.classname=oracle.jdbc.OracleDriver
db.url=jdbc:oracle:thin:@localhost:1521:xe
db.username=system
db.password=123456
```

2) board_mapper.xml

[WebContent-WEB-INF-mapper폴더]

```
<?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">

</mapper>
```

3) servlet-context.xml

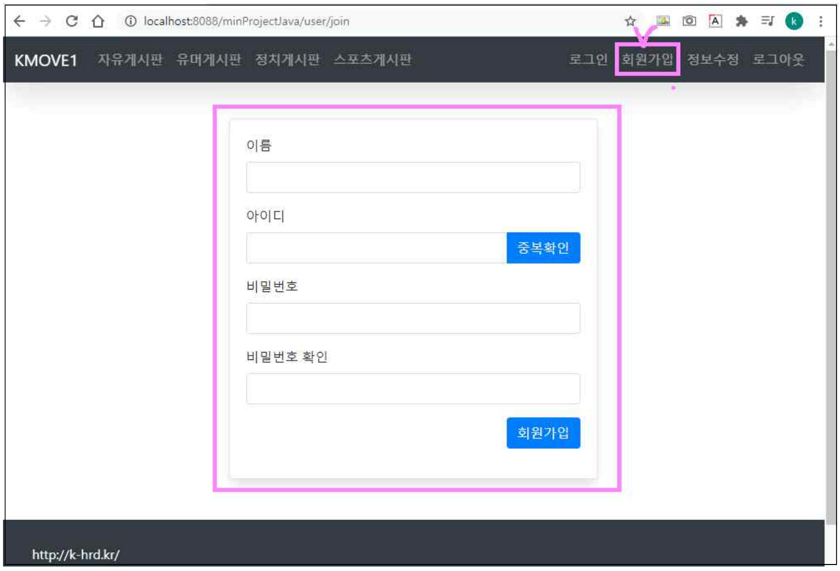
[WebContent-WEB-INF-config폴더]

```
<!--1) properties파일의 내용을 사용할 수 있도록 Bean를 정의 -->
        <beans:bean class='org.springframework.beans.factory.config.PropertyPlaceholderConfigurer'>
                <beans:property name="location">
                        <beans:value>/WEB-INF/properties/db.properties</beans:value>
                </beans:property>
        </beans:bean>

<!--2) -->
        <beans:bean class='org.apache.commons.dbcp2.BasicDataSource' id='basic_data_source'>
                <beans:property name='driverClassName' value='${db.classname}'/>
                <beans:property name="url" value="${db.url}"/>
                <beans:property name="username" value='${db.username}'/>
                <beans:property name="password" value='${db.password}'/>
        </beans:bean>

<!--3) -->
        <beans:bean class='org.mybatis.spring.SqlSessionFactoryBean' id='sqlSession'>
```

```
                <beans:property name="dataSource" ref='basic_data_source'/>
                <beans:property name="mapperLocations" value='/WEB-INF/mapper/*.xml'/>
        </beans:bean>

<!--4) -->
        <beans:bean class='org.mybatis.spring.SqlSessionTemplate' id='sqlSessionTemplate'>
                <beans:constructor-arg index='0' ref='sqlSession'/>
        </beans:bean>
```

2. 회원가입

(1) 회원가입-저장처리



1)-1 UserMapper.java -(mapper에서 쿼리문 만들기) -(Java방식)

[kr.co.korea.mapper]

```
package kr.co.korea.mapper;

import org.apache.ibatis.annotations.Insert;
import org.apache.ibatis.annotations.Select;

import kr.co.korea.beans.UserBean;

public interface UserMapper {

        @Select("select user_name "+
                        "from user_table "+
                        "where user_id= #{user_id}")
         String checkUserIdExist(String user_id);

//(1) 회원가입-저장처리(insert into)
        @Insert("insert into user_table (user_idx, user_name, user_id, user_pw) " +
                        "values (user_seq.nextval, #{user_name}, #{user_id}, #{user_pw})")
        void addUserInfo(UserBean joinUserBean);

}
```

1)-2 user_mapper.xml -(mapper에서 쿼리문 만들기) -(xml방식)

[WebContent-WEB-INF-mapper폴더]

```
<?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="user">
        <select id="checkUserIdExist" parameterType="java.lang.String" resultType="java.lang.String">
                <![CDATA[
                        select user_name
                        from user_table
                        where user_id = #{user_id}
                ]]>
        </select>

<!--(1) mapper  -->
        <insert id="addUserInfo" parameterType="kr.co.softcampus.beans.UserBean">
                <![CDATA[
                        insert into user_table (user_idx, user_name, user_id, user_pw)
                        values (user_seq.nextval, #{user_name}, #{user_id}, #{user_pw})
                ]]>
        </insert>
</mapper>
```

2)-1 UserDao.java -(dao에서 mapper의 addUserInfo() 정의해두기) -(Java방식)

[kr.co.korea.dao]

```
@Repository
public class UserDao {

        @Autowired
        private UserMapper userMapper;
        public String checkUserIdExist(String user_id) {
                return userMapper.checkUserIdExist(user_id);
        }
//(2) mapper의 addUserInfor 호출하도록 addUserInfo()메소드 정의
        public void addUserInfo(UserBean joinUserBean) {
                userMapper.addUserInfo(joinUserBean); //mapper의 addUserInfor호출
        }
}
```

2)-2 UserDao.java -(dao에서 mapper.xml파일내의 "user.addUserInfo" 삽입명령 호출하도록 addUserInfor()메소드 정의해두기) -(xml방식)

[kr.co.korea.dao]

```
package kr.co.korea.dao;

import org.mybatis.spring.SqlSessionTemplate;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;

import kr.co.korea.beans.UserBean;
@Repository
public class UserDao {

        @Autowired
        private SqlSessionTemplate sqlSessionTemplate;

        public String checkUserIdExist(String user_id) {
                return sqlSessionTemplate.selectOne("user.checkUserIdExist", user_id);
        }
//(2) mapper의 "user.addUserInfo" 삽입명령을 호출하도록 addUserInfor()메소드 정의
        public void addUserInfo(UserBean joinUserBean) {
                sqlSessionTemplate.insert("user.addUserInfo", joinUserBean);
        }
}
```

3) UserService.java -(service에서 dao의 addUserInfor() 정의해두기) -(java방식과 xml방식 동일)

[kr.co.korea.service]

```
@Service
public class UserService {

        @Autowired
        private UserDao userDao;
        public boolean checkuserIdExist(String user_id) {
                String user_name=userDao.checkUserIdExist(user_id);
                if(user_name==null) {
                        return true;
                } else {
                        return false;
                }
        }
//(3) UserDao의 addUserInfo() 호출하도록 정의
	public void addUserInfo(UserBean joinUserBean) {
		userDao.addUserInfo(joinUserBean);	//UserDao의 addUserInfo()호출
	}

}
```

4) UserController.java -(service의 addUserInfor() 호출하기) -(java방식과 xml방식 동일)

[kr.co.korea.controller]

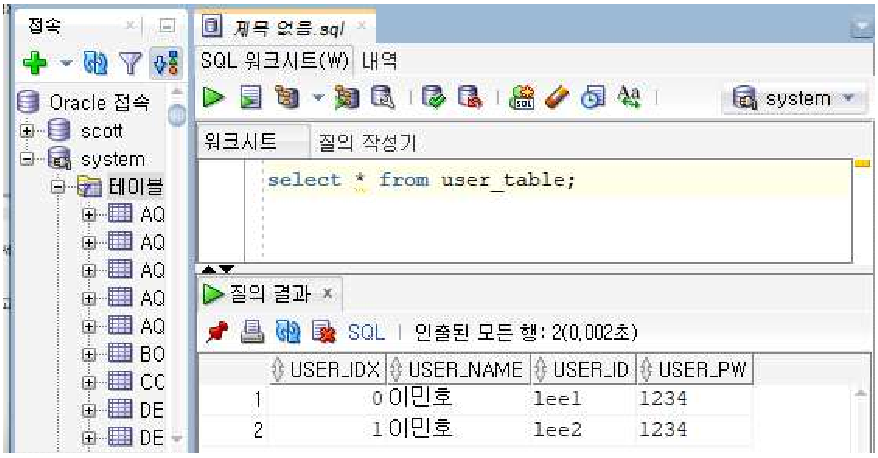
```
@Controller
@RequestMapping("/user")
public class UserController {

//(4) userService 객체로 주입
	@Autowired
	private UserService userService;

	@GetMapping("/login")
	public String login() {
		return "user/login";
	}

//가입창부터 가입완료까지  ----------------------------------
	@GetMapping("/join")
	public String join(@ModelAttribute("joinUserBean") UserBean joinUserBean) {
		return "user/join";	//가입창 페이지
	}

	@PostMapping("/join_pro")	//가입창 페이지 -> “회원가입”버튼 클릭시 처리부분
	public String join_pro(@Valid @ModelAttribute("joinUserBean") UserBean joinUserBean, BindingResult
result) {
		if(result.hasErrors()) {
			return "user/join";
		}

		//(4) 객체를 통해 service의 addUserInfo()메소드 호출
		userService.addUserInfo(joinUserBean);

		return "user/join_success";	//가입완료 메시지 페이지
	}
//----------------------------------

	@GetMapping("modify")
	public String modify() {
		return "user/modify";
	}

	@GetMapping("logout")
	public String logout() {
		return "user/logout";
	}

	@InitBinder
	public void initBinder(WebDataBinder binder) {
		UserValidator validator1 = new UserValidator();
		binder.addValidators(validator1);
	}

}
```

5) join_success.jsp -(가입완료 메시지) -(java방식과 xml방식 동일)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:set var="root" value="${pageContext.request.contextPath }/"/>
<script>
```

```
        alert('가입이 완료되었습니다')
        location.href = "${root}user/login"
</script>
```

6) 저장된 결과 확인



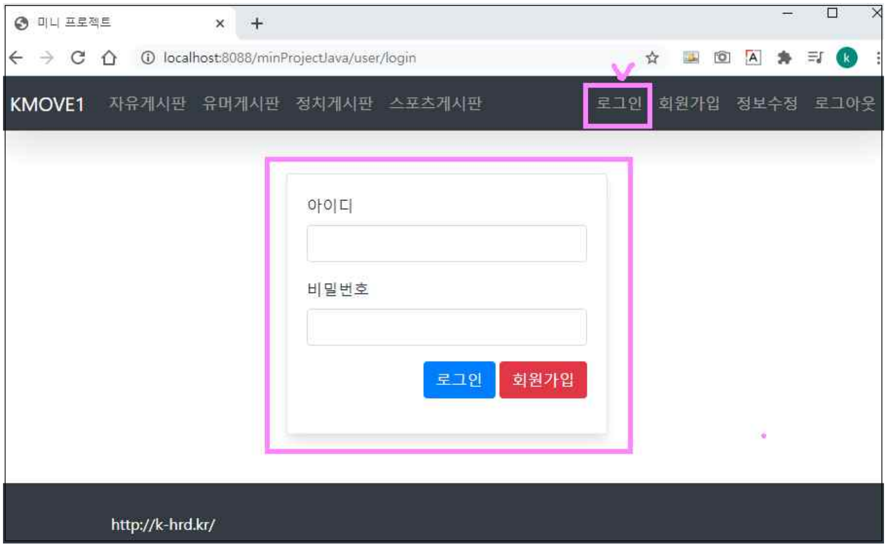
```
select * from user_table;
```

인출된 모든 행: 2(0.002초)

	USER_IDX	USER_NAME	USER_ID	USER_PW
1	0	이민호	lee1	1234
2	1	이민호	lee2	1234

3. 로그인

(1) 로그인 -유효성 검사 (Java방식)



1) UserBean.java (XML방식 동일)

```
public class UserBean {
        private int user_idx;

        @Size(min=2, max=4)
        @Pattern(regexp = "[가-힣]*")
        private String user_name;

        @Size(min=4, max=20)
        @Pattern(regexp = "[a-zA-Z0-9]*")
        private String user_id;

        @Size(min=4, max=20)
        @Pattern(regexp = "[a-zA-Z0-9]*")
        private String user_pw;
```

```
	@Size(min=4, max=20)
	@Pattern(regexp = "[a-zA-Z0-9]*")
	private String user_pw2;

	private boolean userIdExist;

	//로그인 되었는지 로그인 성공시 true저장
	private boolean userLogin;

	public UserBean() {
		this.userIdExist = false;

		this.userLogin=false;
	}

	public int getUser_idx() {
		return user_idx;
	}
	public void setUser_idx(int user_idx) {
		this.user_idx = user_idx;
	}
	//이하생략

	public boolean isUserLogin() {
		return userLogin;
	}

	public void setUserLogin(boolean userLogin) {
		this.userLogin = userLogin;
	}

}
```

2)-1 RootAppContext.java -(UserBean 빈객체 만들어두기)

```
//빈객체 만들어두기
@Configuration
public class RootAppContext {

	@Bean("loginUserBean")
	@SessionScope
	public UserBean loginUserBean() {
		return new UserBean();
	}

}
```

```
//주입되는지 테스트하기
@Controller
public class HomeController {

//	@Resource(name="loginUserBean")
//	private UserBean loginUserBean;
	@RequestMapping(value="/",
method=RequestMethod.GET)
	public String home() {
//		System.out.println(loginUserBean);
		return "redirect:main";
	}
}
```

2)-2 root-context.xml -(UserBean 빈객체 만들어두기)

```
//빈객체 만들어두기
<?xml version="1.0" encoding="UTF-8"?>
<beans
xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

	<bean class="kr.co.korea.beans.UserBean"
id="loginUserBean" scope="session"/>

</beans>
```

```
//주입되는지 테스트하기
@Controller
public class HomeController {

//	@Resource(name="loginUserBean")
//	@Lazy
//	private UserBean loginUserBean;

	@RequestMapping(value="/",
method=RequestMethod.GET)
	public String home() {
//		System.out.println(loginUserBean);
		return "redirect:main";
	}
}
```

3) login.jsp -(form 태그라이브러리 활용) (XML방식 동일)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>

<c:set var='root' value="${pageContext.request.contextPath }/" />
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>웹 프로젝트</title>
<!-- Bootstrap CDN -->
<link rel="stylesheet"
    href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css">
<script
    src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script
    src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js"></script>
<script
    src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js"></script>
</head>
<body>

    <c:import url="/WEB-INF/views/include/top_menu.jsp" />

<div class="container" style="margin-top: 100px">
<div class="row">
    <div class="col-sm-3"></div>
    <div class="col-sm-6">
        <div class="card shadow">
            <div class="card-body">

            <div class="alert alert-danger">
                <h3>로그인 실패</h3>
                <p>아이디 비밀번호를 확인해주세요</p>
            </div>

        <form:form action="${root }user/login_pro" method='post'
            modelAttribute="tempLoginUserBean">
            <div class="form-group">
                <form:label path="user_id">아이디</form:label>
                <form:input path="user_id" class='form-control' />
                <form:errors path='user_id' style='color:red' />
            </div>
            <div class="form-group">
                <form:label path="user_pw">비밀번호</form:label>
                <form:password path="user_pw" class='form-control' />
                <form:errors path='user_pw' style='color:red' />
            </div>
            <div class="form-group text-right">
                <form:button class='btn btn-primary'>로그인</form:button>
                <a href="${root }user/join" class="btn btn-danger">회원가입</a>
            </div>
        </form:form>

            </div>
        </div>
    </div>
        <div class="col-sm-3"></div>
</div>
</div>

<c:import url="/WEB-INF/views/include/bottom_info.jsp" />

</body>
</html>
```

4) UserController.java -(XML방식 동일)

```
@Controller
@RequestMapping("/user")
public class UserController {

//userService 객체로 주입
```

```
        @Autowired
        private UserService userService;

        @GetMapping("/login")
        public String login(@ModelAttribute("tempLoginUserBean") UserBean tempLoginUserBean) {
                return "user/login";
        }



        @PostMapping("/login_pro")
        public String login_pro(@Valid @ModelAttribute("tempLoginUserBean") UserBean tempLoginUserBean,
BindingResult result) {

                if(result.hasErrors()) {
                        return "user/login";
                }
                return "user/login_success";
        }


        @GetMapping("/join")
        public String join(@ModelAttribute("joinUserBean") UserBean joinUserBean) {
                return "user/join";
        }

        @PostMapping("/join_pro")
        public String join_pro(@Valid @ModelAttribute("joinUserBean") UserBean joinUserBean, BindingResult
result) {
                if(result.hasErrors()) {
                        return "user/join";
                }

                //객체를 통해 service의 addUserInfo()메소드 호출
                userService.addUserInfo(joinUserBean);

                return "user/join_success";
        }

        @GetMapping("modify")
        public String modify() {
                return "user/modify";
        }

        @GetMapping("logout")
        public String logout() {
                return "user/logout";
        }

        @InitBinder
        public void initBinder(WebDataBinder binder) {
                UserValidator validator1 = new UserValidator();
                binder.addValidators(validator1);
        }

}
```

5) error_message.properties

```
Size.joinUserBean.user_name =사용자 이름은 2~4글자여야 합니다.
Size.joinUserBean.user_id = 사용자 아이디는 4~20글자여야 합니다.
Size.joinUserBean.user_pw = 비밀번호는 4~20글자여야 합니다.
Size.joinUserBean.user_pw2 = 비밀번호는 4~20글자여야 합니다.
Pattern.joinUserBean.user_name= 사용자 이름은 한글만 허용합니다.
Pattern.joinUserBean.user_id = 사용자 아이디는 영문자대소문자, 숫자만 허용합니다.
Pattern.joinUserBean.user_pw = 비밀번호는 영문대소문자, 숫자만 허용합니다.
Pattern.joinUserBean.user_pw2 = 비밀번호는 영문대소문자, 숫자만 허용합니다


NotEquals.joinUserBean.user_pw = 비밀번호가 일치하지 않습니다
NotEquals.joinUserBean.user_pw2 = 비밀번호가 일치하지 않습니다

DontCheckUserIdExist.joinUserBean.user_id = 중복확인을 해주세요
```

```
Size.tempLoginUserBean.user_name =사용자 이름은 2~4글자여야 합니다.
Size.tempLoginUserBean.user_id = 사용자 아이디는 4~20글자여야 합니다.
Pattern.tempLoginUserBean.user_id = 사용자 아이디는 영문자대소문자, 숫자만 허용합니다.
Pattern.tempLoginUserBean.user_pw = 비밀번호는 영문대소문자, 숫자만 허용합니다.
```

6) UserValidator.java (UserBean 클래스를 joinUserBean, tempLoginUserBean 2개의 이름으로 사용하므로 validate메소드의 유효성 검사를 모두 하게 된다. joinUserBean일때만 validate메소드 호출하도록 조건지정)

```java
package kr.co.korea.validator;

import org.springframework.validation.Errors;
import org.springframework.validation.Validator;

import kr.co.korea.beans.UserBean;

public class UserValidator implements Validator{

	@Override
	public boolean supports(Class<?> clazz) {
		// TODO Auto-generated method stub
		return UserBean.class.isAssignableFrom(clazz);
	}

	@Override
	public void validate(Object target, Errors errors) {
		// TODO Auto-generated method stub
		UserBean userBean = (UserBean)target;

		String beanName = errors.getObjectName();

		if(beanName.equals("joinUserBean")) {
			if(userBean.getUser_pw().equals(userBean.getUser_pw2()) == false) {
				errors.rejectValue("user_pw", "NotEquals");
				errors.rejectValue("user_pw2", "NotEquals");
			}

			if(userBean.isUserIdExist() == false) {
				errors.rejectValue("user_id", "DontCheckUserIdExist");
			}
		}

	}

}
```

7) login_success.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix='c' uri="http://java.sun.com/jsp/jstl/core" %>
<c:set var='root' value='${pageContext.request.contextPath }/'/>
<script>
	alert('로그인 되었습니다')
	location.href = '${root}main'
</script>
```

(2) 로그인 정보와 일치하는 sql쿼리 조회

select user_idx, user_name from user_table where user_id='choi1' and user_pw='123456';

1)-1 UserMapper.java -(mapper에서 쿼리문 만들기) -(Java방식)

[kr.co.korea.mapper]패키지

```java
package kr.co.korea.mapper;
```

```
import org.apache.ibatis.annotations.Insert;
import org.apache.ibatis.annotations.Select;

import kr.co.korea.beans.UserBean;

public interface UserMapper {

        @Select("select user_name "+
                        "from user_table "+
                        "where user_id= #{user_id}")
         String checkUserIdExist(String user_id);

        @Insert("insert into user_table (user_idx, user_name, user_id, user_pw) " +
                        "values (user_seq.nextval, #{user_name}, #{user_id}, #{user_pw})")
        void addUserInfo(UserBean joinUserBean);
// 로그인 (아이디와 패스워드 일치할 경우 idx, name 조회)
        @Select("select user_idx, user_name " +
                        "from user_table " +
                        "where user_id=#{user_id} and user_pw=#{user_pw}")
        UserBean getLoginUserInfo(UserBean tempLoginUserBean);

}
```

1)-2 user_mapper.xml -(mapper에서 쿼리문 만들기) -(xml방식)

[WebContent-WEB-INF-mapper]폴더

```
<?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="user">
        <select id="checkUserIdExist" parameterType="java.lang.String" resultType="java.lang.String">
                <![CDATA[
                        select user_name
                        from user_table
                        where user_id = #{user_id}
                ]]>
        </select>

        <insert id="addUserInfo" parameterType="kr.co.korea.beans.UserBean">
                <![CDATA[
                        insert into user_table (user_idx, user_name, user_id, user_pw)
                        values (user_seq.nextval, #{user_name}, #{user_id}, #{user_pw})
                ]]>
        </insert>

<!--(1) mapper  아이디,비번이 일치하는 idx,name 조회-->
        <select id="getLoginUserInfo" parameterType="kr.co.softcampus.beans.UserBean"
resultType="kr.co.softcampus.beans.UserBean">
                <![CDATA[
                        select user_idx, user_name
                        from user_table
                        where user_id=#{user_id} and user_pw=#{user_pw}
                ]]>
        </select>

</mapper>
```

2)-1 UserDao.java -(dao에서 mapper의 getLoginUserInfo() 정의해두기) -(Java방식)

[kr.co.korea.dao]

```
@Repository
public class UserDao {

        @Autowired
        private UserMapper userMapper;
        public String checkUserIdExist(String user_id) {
                return userMapper.checkUserIdExist(user_id);
        }


        public void addUserInfo(UserBean joinUserBean) {
                userMapper.addUserInfo(joinUserBean); /
        }
//mapper의 getLoginUserInfo 호출하도록 정의
	public UserBean getLoginUserInfo(UserBean tempLoginUserBean) {
		return userMapper.getLoginUserInfo(tempLoginUserBean); //mapper의 getLoginUserInfo호출
	}

}
```

2)-2 UserDao.java -(mapper의 'user.getLoginUserInfo' 조회명령 호출하도록 getLoginUserInfo() 정의해두기) -(xml방식)

[kr.co.korea.dao]

```
@Repository
public class UserDao {

	@Autowired
	private SqlSessionTemplate sqlSessionTemplate;

	public String checkUserIdExist(String user_id) {
		return sqlSessionTemplate.selectOne("user.checkUserIdExist", user_id);
	}

	public void addUserInfo(UserBean joinUserBean) {
		sqlSessionTemplate.insert("user.addUserInfo", joinUserBean);
	}

	public UserBean getLoginUserInfo(UserBean tempLoginUserBean) {
		return sqlSessionTemplate.selectOne("user.getLoginUserInfo", tempLoginUserBean);
	}

}
```

3)-1 UserService.java -(service에서 dao의 getLoginUserInfo() 정의해두기) -(Java방식)

[kr.co.korea.service]

```
@Service
public class UserService {


	@Autowired
	private UserDao userDao;

	@Resource(name = "loginUserBean")
	private UserBean loginUserBean;


	public boolean checkuserIdExist(String user_id) {
		String user_name=userDao.checkUserIdExist(user_id);
		if(user_name==null) {
			return true;
		} else {
			return false;
		}
	}

	public void addUserInfo(UserBean joinUserBean) {
		userDao.addUserInfo(joinUserBean);
	}

//UserDao의 getLoginUserInfo() 호출하도록 정의
	public void getLoginUserInfo(UserBean tempLoginUserBean) {

		UserBean tempLoginUserBean2 = userDao.getLoginUserInfo(tempLoginUserBean); //호출

		if(tempLoginUserBean2 != null) { //db에서 select된게 있으면
			loginUserBean.setUser_idx(tempLoginUserBean2.getUser_idx());
			loginUserBean.setUser_name(tempLoginUserBean2.getUser_name());
			loginUserBean.setUserLogin(true);
		}
```

```
        }
}
```

3)-2 UserService.java -(service에서 dao의 getLoginUserInfo() 정의해두기) -(xml방식)
[kr.co.korea.service]

```
@Service
public class UserService {

        @Autowired
        private UserDao userDao;

        @Resource(name = "loginUserBean")
        @Lazy
        private UserBean loginUserBean;

        public boolean checkuserIdExist(String user_id) {
                String user_name=userDao.checkUserIdExist(user_id);
                if(user_name==null) {
                        return true;
                } else {
                        return false;
                }
        }

        public void addUserInfo(UserBean joinUserBean) {
                userDao.addUserInfo(joinUserBean);      //UserDao의 addUserInfo()호출
        }

//UserDao의 getLoginUserInfo() 호출하도록 정의
        public void getLoginUserInfo(UserBean tempLoginUserBean) {

                UserBean tempLoginUserBean2 = userDao.getLoginUserInfo(tempLoginUserBean);  //호출

                if(tempLoginUserBean2 != null) { //db에서 select된게 있으면
                        loginUserBean.setUser_idx(tempLoginUserBean2.getUser_idx());
                        loginUserBean.setUser_name(tempLoginUserBean2.getUser_name());
                        loginUserBean.setUserLogin(true);
                }
        }
}
```

4) UserController.java -(service의 getLoginUserInfo() 호출하기) -java방식과 xml방식 동일
[kr.co.korea.controller]

```
@Controller
@RequestMapping("/user")
public class UserController {

        @Autowired
        private UserService userService;

        @Resource(name = "loginUserBean")
        private UserBean loginUserBean;

        @GetMapping("/login")
        public String login(@ModelAttribute("tempLoginUserBean") UserBean tempLoginUserBean) {
                return "user/login";
        }

        @PostMapping("/login_pro")
        public String login_pro(@Valid @ModelAttribute("tempLoginUserBean") UserBean tempLoginUserBean,
BindingResult result) {

                if(result.hasErrors()) {
                        return "user/login";
        }
        userService.getLoginUserInfo(tempLoginUserBean);   //db를 통해 조회된 idx,name select하기

        if(loginUserBean.isUserLogin() == true) {
            return "user/login_success";
        } else {
            return "user/login_fail";
        }
    }

    @GetMapping("/join")
    public String join(@ModelAttribute("joinUserBean") UserBean joinUserBean) {
        return "user/join";
    }

    @PostMapping("/join_pro")
    public String join_pro(@Valid @ModelAttribute("joinUserBean") UserBean joinUserBean, BindingResult
result) {
        if(result.hasErrors()) {
            return "user/join";
        }

        //객체를 통해 service의 addUserInfo()메소드 호출
        userService.addUserInfo(joinUserBean);

        return "user/join_success";
    }

    @GetMapping("modify")
    public String modify() {
        return "user/modify";
    }

    @GetMapping("logout")
    public String logout() {
        return "user/logout";
    }

    @InitBinder
    public void initBinder(WebDataBinder binder) {
        UserValidator validator1 = new UserValidator();
        binder.addValidators(validator1);
    }

}
```

5) login_fail.jsp -(가입완료 메시지) -java방식과 xml방식 동일

실패시 로그인창으로 다시 이동

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:set var='root' value='${pageContext.request.contextPath}/'/>
<script>
    alert('로그인에 실패하였습니다')
    location.href = '${root}user/login'
</script>
```

(3) 로그인 실패와 성공 표현

1) UserController.java

```
@RequestMapping("/user")
public class UserController {


    @Autowired
    private UserService userService;
```

```
        @Resource(name = "loginUserBean")
        private UserBean loginUserBean;

        @GetMapping("/login")
        public String login(@ModelAttribute("tempLoginUserBean") UserBean tempLoginUserBean,
                        @RequestParam(value = "logcheck", defaultValue = "true") boolean logcheck,
                        Model model) {

                model.addAttribute("logcheck", logcheck);  //logcheck

                return "user/login";
        }
//이하 생략
```

2) login.jsp

```
<body>

        <c:import url="/WEB-INF/views/include/top_menu.jsp" />

<div class="container" style="margin-top: 100px">
<div class="row">
        <div class="col-sm-3"></div>
        <div class="col-sm-6">
                <div class="card shadow">
                        <div class="card-body">
                <c:if test="${logcheck == false }">
                        <div class="alert alert-danger">
                                <h3>로그인 실패</h3>
                                <p>아이디 비밀번호를 확인해주세요</p>
                        </div>
                </c:if>
                <form:form action="${root }user/login_pro" method='post'
                        modelAttribute="tempLoginUserBean">
                        <div class="form-group">
                                <form:label path="user_id">아이디</form:label>
                                <form:input path="user_id" class='form-control' />
                                <form:errors path='user_id' style='color:red' />
                        </div>
                        <div class="form-group">
                                <form:label path="user_pw">비밀번호</form:label>
                                <form:password path="user_pw" class='form-control' />
                                <form:errors path='user_pw' style='color:red' />
                        </div>
                        <div class="form-group text-right">
                                <form:button class='btn btn-primary'>로그인</form:button>
                                <a href="${root }user/join" class="btn btn-danger">회원가입</a>
                        </div>
                </form:form>
```

3) login_fail.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:set var='root' value='${pageContext.request.contextPath}/'/>
<script>
        alert('로그인에 실패하였습니다')
        location.href = '${root}user/login?logcheck=false'
</script>
```

4. 로그아웃

(1) 로그인된 경우 로그아웃 글자만 표현

1) ServletAppContext.java

```
@Resource(name = "loginUserBean")
private UserBean loginUserBean;
```

```
@Override
public void addInterceptors(InterceptorRegistry registry) {
        // TODO Auto-generated method stub
        WebMvcConfigurer.super.addInterceptors(registry);
        TopMenuInterceptor topMenuIntercepter=new TopMenuInterceptor(TopMenuService, loginUserBean);

        InterceptorRegistration reg1=registry.addInterceptor(topMenuIntercepter);
        reg1.addPathPatterns("/**");
}
```

2)-1 TopMenuInterceptor.java -(java방식)

```
public class TopMenuInterceptor implements HandlerInterceptor{

        private TopMenuService topMenuService;
        private UserBean loginUserBean;

        public TopMenuInterceptor(TopMenuService topMenuService, UserBean loginUserBean) {
                this.topMenuService=topMenuService;
                this.loginUserBean=loginUserBean;
                // TODO Auto-generated constructor stub
        }

        @Override
        public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object
handler)
                        throws Exception {
                // TODO Auto-generated method stub
                List<BoardInfoBean> topMenuList=topMenuService.getTopMenuList();
                request.setAttribute("topMenuList", topMenuList);
                request.setAttribute("loginUserBean", loginUserBean);

                return true;
        }
}
```

2)-2 TopMenuInterceptor.java -(xml방식)

```
public class TopMenuInterceptor implements HandlerInterceptor{

        @Autowired
        private TopMenuService topMenuService;

        @Resource(name="loginUserBean")
        @Lazy
        private UserBean loginUserBean;


        @Override
        public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object
handler)
                        throws Exception {
                // TODO Auto-generated method stub
                List<BoardInfoBean> topMenuList = topMenuService.getTopMenuList();
                request.setAttribute("topMenuList", topMenuList);
                request.setAttribute("loginUserBean", loginUserBean);
                return true;
        }
}
```

3) top_menu.jsp -java방식과 xml방식 동일

```
<nav class="navbar navbar-expand-md bg-dark navbar-dark fixed-top shadow-lg">
        <a class="navbar-brand" href="${root }main">KMOVE1</a>
<button class="navbar-toggler" type="button" data-toggle="collapse"
       data-target="#navMenu">
        <span class="navbar-toggler-icon"></span>
</button>
        <ul class="navbar-nav ml-auto">
        <c:choose>
                <c:when test="${loginUserBean.userLogin == true }">  //userLogin값이 true이면 로그인상태
                        <li class="nav-item">
                                <a href="${root }user/modify" class="nav-link">정보수정</a>
                        </li>
                        <li class="nav-item">
                                <a href="${root }user/logout" class="nav-link">로그아웃</a>
                        </li>
                </c:when>
                <c:otherwise>                  //userLogin값이 false 이면 로그아웃상태
                        <li class="nav-item">
                                <a href="${root }user/login" class="nav-link">로그인</a>
                        </li>
                        <li class="nav-item">
                                <a href="${root }user/join" class="nav-link">회원가입</a>
                        </li>
                </c:otherwise>
        </c:choose>

                </ul>
        </div>
</nav>
```

4) UserController.java -java방식과 xml방식 동일

```
        @GetMapping("logout")
        public String logout() {

                loginUserBean.setUserLogin(false); //userLogin변수에 false값을 세트

                return "user/logout";
        }
```

(2) 로그인된 경우에만 정보 확인 가능 (url에 주소 입력할 경우 로그인창으로 이동)

1)-1 CheckLoginInterceptor.java -(java방식)

```
public class CheckLoginInterceptor implements HandlerInterceptor{

        private UserBean loginUserBean;

        public CheckLoginInterceptor(UserBean loginUserBean) {
                // TODO Auto-generated constructor stub
                this.loginUserBean = loginUserBean;
        }

        @Override
        public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object
handler)
                        throws Exception {
                // TODO Auto-generated method stub
                if(loginUserBean.isUserLogin() == false) {
                        String contextPath = request.getContextPath();
                        response.sendRedirect(contextPath + "/user/not_login");
                        return false;
                }
                return true;
        }
}
```

1)-2 CheckLoginInterceptor.java -(xml방식)

```
public class CheckLoginInterceptor implements HandlerInterceptor{

//      private UserBean loginUserBean;
//
//      public CheckLoginInterceptor(UserBean loginUserBean) {
//              // TODO Auto-generated constructor stub
//              this.loginUserBean = loginUserBean;
//      }
```

```
	@Resource(name="loginUserBean")
	@Lazy
	private UserBean loginUserBean;

	@Override
	public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object
handler)
			throws Exception {
		// TODO Auto-generated method stub
		if(loginUserBean.isUserLogin() == false) {
			String contextPath = request.getContextPath();
			response.sendRedirect(contextPath + "/user/not_login");
			return false;
		}
		return true;
	}
}
```

2)-1 ServletAppContext.java   -(java방식)

```
	@Override
	public void addInterceptors(InterceptorRegistry registry) {
		// TODO Auto-generated method stub
		WebMvcConfigurer.super.addInterceptors(registry);
		TopMenuInterceptor topMenuIntercepter = new TopMenuInterceptor(TopMenuService,
loginUserBean);

		InterceptorRegistration reg1 = registry.addInterceptor(topMenuIntercepter);
		reg1.addPathPatterns("/**");

		CheckLoginInterceptor checkLoginInterceptor = new CheckLoginInterceptor(loginUserBean);
		InterceptorRegistration reg2 = registry.addInterceptor(checkLoginInterceptor);
		reg2.addPathPatterns("/user/modify", "/user/logout", "/board/*");
		reg2.excludePathPatterns("/board/main");

	}
```

2)-2 servlet-context.xml -(xml방식)

```
	<interceptors>
		<interceptor>
			<mapping path="/**"/>
			<beans:bean class='kr.co.korea.interceptor.TopMenuInterceptor'/>
		</interceptor>

		<interceptor>
			<mapping path="/user/modify"/>
			<mapping path="/user/logout"/>
			<mapping path="/board/*"/>
			<exclude-mapping path="/board/main"/>
			<beans:bean class='kr.co.korea.interceptor.CheckLoginInterceptor'/>
		</interceptor>

	</interceptors>
```

3) UserController.java

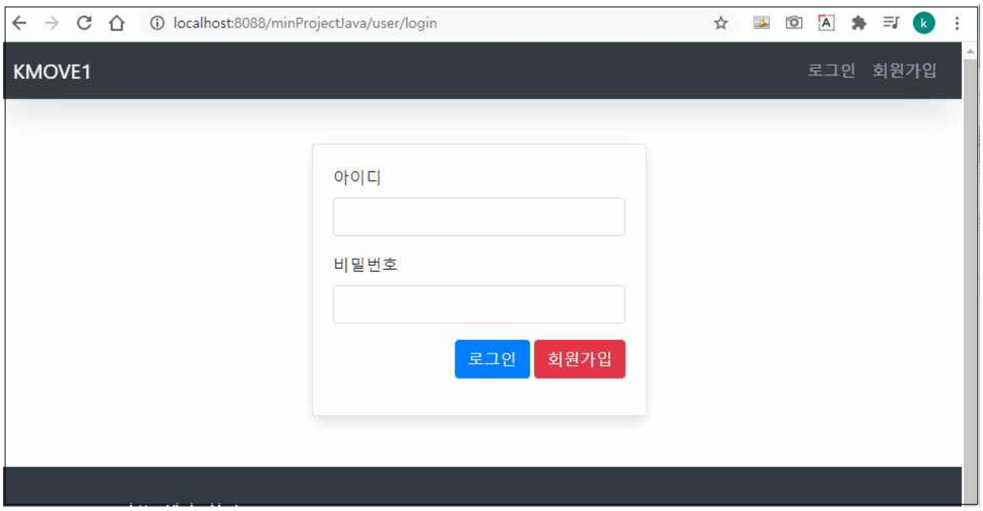
```
	@GetMapping("/not_login")
	public String not_login() {
		return "user/not_login";
	}
```

4) not_login.jsp

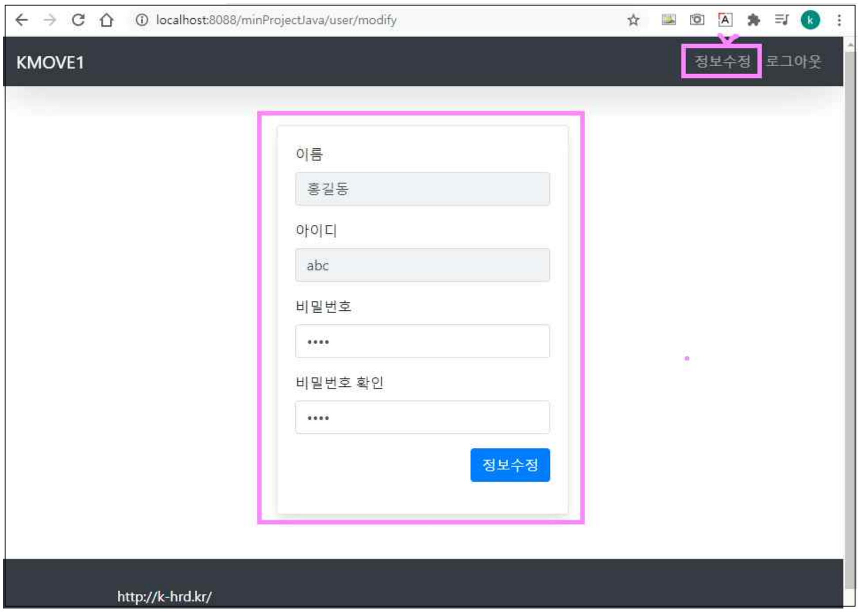
```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:set var="root" value="${pageContext.request.contextPath }/"/>
<script>
```

```
            alert('로그인 해주세요')
            location.href='${root}user/login'
</script>
```

5) 실습시 :http://localhost:8088/webProjectJava/user/modify 입력시 user/login으로 이동하도록


localhost:8088/minProjectJava/user/login

5. 정보수정


localhost:8088/minProjectJava/user/modify

이름: 홍길동

아이디: abc

비밀번호: ····

비밀번호 확인: ····

(1) 로그인 정보에 대한 - 정보수정

1) modify.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
```

```
<c:set var='root' value='${pageContext.request.contextPath }/'/>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>웹 프로젝트</title>
<!-- Bootstrap CDN -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js"></script>

</head>
<body>

<c:import url="/WEB-INF/views/include/top_menu.jsp"/>

<div class="container" style="margin-top:100px">
<div class="row">
        <div class="col-sm-3"></div>
        <div class="col-sm-6">
                <div class="card shadow">
                        <div class="card-body">
```

```
                <form:form action='${root }user/modify_pro' method='post'
modelAttribute="modifyUserBean">
                        <div class="form-group">
                                <form:label path="user_name">이름</form:label>
                                <form:input path="user_name" class='form-control' readonly="true"/>
                        </div>
                        <div class="form-group">
                                <form:label path="user_id">아이디</form:label>
                                <form:input path="user_id" class='form-control' readonly="true"/>
                        </div>
                        <div class="form-group">
                                <form:label path="user_pw">비밀번호</form:label>
                                <form:password path="user_pw" class='form-control'/>
                                <form:errors path='user_pw' style='color:red'/>
                        </div>
                        <div class="form-group">
                                <form:label path="user_pw2">비밀번호 확인</form:label>
                                <form:password path="user_pw2" class='form-control'/>
                                <form:errors path='user_pw2' style='color:red'/>
                        </div>
                        <div class="form-group">
                                <div class="text-right">
                                        <form:button class='btn btn-primary'>정보수정</form:button>
                                </div>
                        </div>
                </form:form>
```

```
                                </div>
                        </div>
                </div>
                <div class="col-sm-3"></div>
        </div>
</div>

<c:import url="/WEB-INF/views/include/bottom_info.jsp"/>

</body>
</html>
```

2) UserController.java

```
@GetMapping("/modify")
public String modify(@ModelAttribute("modifyUserBean") UserBean modifyUserBean) {
        return "user/modify";
}
```

(3) db에서 로그인한 정보 가져오기

1)-1 UserMapper.java -(mapper에서 쿼리문 만들기) -(Java방식)

[kr.co.korea.mapper]

```
// 로그인한 사람의 정보 가져와서 "정보수정" 페이지에서 활용
        @Select("select user_id, user_name " +
                        "from user_table " +
                        "where user_idx = #{user_idx}")
        UserBean getModifyUserInfo(int user_idx);
```

1)-2 user_mapper.xml -(mapper에서 쿼리문 만들기) -(xml방식)

```
<!--(1) mapper  아이디와 일치하는 id,name 조회하여 '정보수정' 페이지에서 활용      -->
        <select id="getModifyUserInfo" parameterType="java.lang.Integer"
resultType="kr.co.softcampus.beans.UserBean">
                <![CDATA[
                        select user_id, user_name
                        from user_table
                        where user_idx = #{user_idx}
                ]]>
        </select>
```

2)-1 UserDao.java -(dao에서 mapper의 getModifyUserInfo() 정의해두기) -(Java방식)

[kr.co.korea.dao]

```
//mapper의 getModifyUserInfo 호출하도록 정의
        public UserBean getModifyUserInfo(int user_idx) {
                return userMapper.getModifyUserInfo(user_idx);
        }
```

2)-2 UserDao.java -(dao에서 mapper.xml파일내의 “user.getModifyUserInfo” 조회하도록 getModifyUserInfo()메소드 정의해두기) -(xml방식)

```
        public UserBean getModifyUserInfo(int user_idx) {
                return sqlSessionTemplate.selectOne("user.getModifyUserInfo", user_idx);
        }
```

3) UserService.java -(service에서 dao의 getModifyUserInfo() 정의해두기) -(java방식과 xml방식 동일)

[kr.co.korea.service]

```
//UserDao의 getModifyUserInfo() 호출하도록 정의
        public void getModifyUserInfo(UserBean modifyUserBean) {
                UserBean tempModifyUserBean = userDao.getModifyUserInfo(loginUserBean.getUser_idx());

                modifyUserBean.setUser_id(tempModifyUserBean.getUser_id());
                modifyUserBean.setUser_name(tempModifyUserBean.getUser_name());
                modifyUserBean.setUser_idx(loginUserBean.getUser_idx());
        }
```

4) UserController.java

```
@GetMapping("/modify")
public String modify(@ModelAttribute("modifyUserBean") UserBean modifyUserBean) {

        userService.getModifyUserInfo(modifyUserBean);

        return "user/modify";
}


@PostMapping("/modify_pro")
        public String modify_pro(@Valid @ModelAttribute("modifyUserBean") UserBean modifyUserBean, BindingResult result) {

                if (result.hasErrors()) {
                        return "user/modify";
                }
                return "user/modify_success";
        }
```

(4) 유효성 검사

1) error_message.properties

```
Size.modifyUserBean.user_pw = 비밀번호는 4 ~ 20글자여야 합니다
Size.modifyUserBean.user_pw2 = 비밀번호는 4 ~ 20글자여야 합니다
Pattern.modifyUserBean.user_pw = 비밀번호는 영문대소문자, 숫자만 허용합니다
Pattern.modifyUserBean.user_pw2 = 비밀번호는 영문대소문자, 숫자만 허용합니다
NotEquals.modifyUserBean.user_pw = 비밀번호가 일치하지 않습니다
NotEquals.modifyUserBean.user_pw2 = 비밀번호가 일치하지 않습니다
```

2) UserValidator.java

```
        @Override
        public void validate(Object target, Errors errors) {
                // TODO Auto-generated method stub
                UserBean userBean = (UserBean)target;

                String beanName = errors.getObjectName();


                if(beanName.equals("joinUserBean") || beanName.equals("modifyUserBean")) {
                        if(userBean.getUser_pw().equals(userBean.getUser_pw2()) == false) {
                                errors.rejectValue("user_pw", "NotEquals");
                                errors.rejectValue("user_pw2", "NotEquals");
                        }
                }

                if(beanName.equals("joinUserBean")) {

                        if(userBean.isUserIdExist() == false) {
                                errors.rejectValue("user_id", "DontCheckUserIdExist");
                        }
                }
        }
```

(5) db에 정보수정

1)-1 UserMapper.java -(Java방식)

```
// id 일치하는 것 찾아 비번 변경
        @Update("update user_table " +
                        "set user_pw = #{user_pw} " +
                        "where user_idx = #{user_idx}")
        void modifyUserInfo(UserBean modifyUserBean);
```

1)-2 user_mapper.xml -(xml방식)

```
        <update id="modifyUserInfo" parameterType="kr.co.softcampus.beans.UserBean">
                <![CDATA[
                        update user_table
                        set user_pw = #{user_pw}
                        where user_idx = #{user_idx}
                ]]>
        </update>
```

2)-1 UserDao.java -(Java방식)

```
        public void modifyUserInfo(UserBean modifyUserBean) {
                userMapper.modifyUserInfo(modifyUserBean);
        }
```

2)-2 UserDao.java -(xml방식)

```
        public void modifyUserInfo(UserBean modifyUserBean) {
                sqlSessionTemplate.update("user.modifyUserInfo", modifyUserBean);
        }
```

3) UserService.java

```
        public void modifyUserInfo(UserBean modifyUserBean) {

                modifyUserBean.setUser_idx(loginUserBean.getUser_idx());

                userDao.modifyUserInfo(modifyUserBean);
        }
```

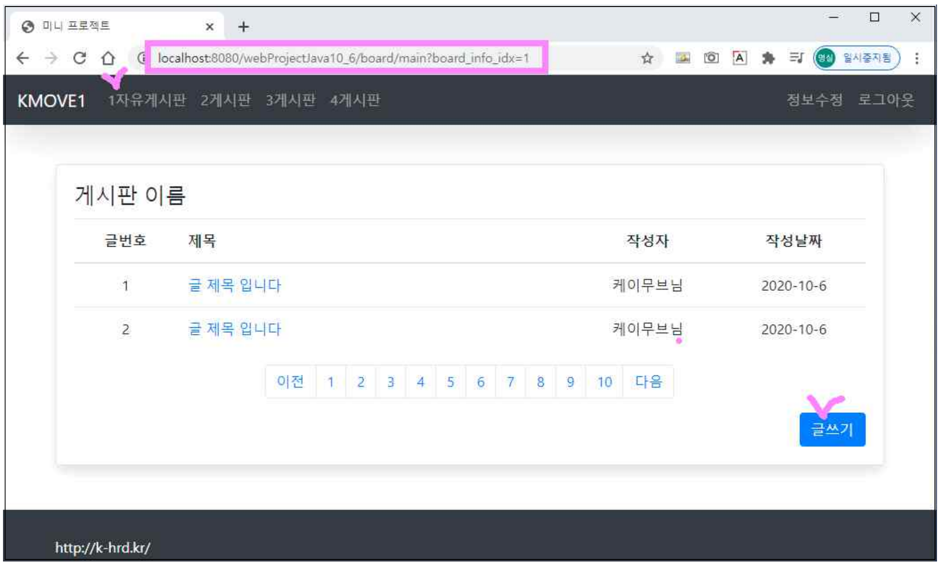
4) UserController.java

```
        @PostMapping("/modify_pro")
        public String modify_pro(@Valid @ModelAttribute("modifyUserBean") UserBean modifyUserBean,
BindingResult result) {

                if (result.hasErrors()) {
                        return "user/modify";
                }

                userService.modifyUserInfo(modifyUserBean);

                return "user/modify_success";
        }
```

5) modify_success.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:set var='root' value='${pageContext.request.contextPath}/'/>
<script>
        alert('수정되었습니다')
        location.href = '${root}user/modify'
</script>
```

6. 글 작성하기



미니 프로젝트

localhost:8080/webProjectJava10_6/board/main?board_info_idx=1

게시판 이름

글번호	제목	작성자	작성날짜
1	글 제목 입니다	케이무브님	2020-10-6
2	글 제목 입니다	케이무브님	2020-10-6

(1) 해당 게시판에 글쓰기

1) main.jsp

```
//          ;
<div class="text-right">
    <a href="${root }board/write?board_info_idx=${board_info_idx}" class="btn btn-primary">글쓰기</a>
</div>
```

2) BoardController.java

```
@GetMapping("/main")
public String main(@RequestParam("board_info_idx") int board_info_idx,
                                        Model model) {

        model.addAttribute("board_info_idx", board_info_idx);

        return "board/main";
}
```

3) board/write.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>

<c:set var="root" value="${pageContext.request.contextPath }/"/>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>웹 프로젝트</title>
<!-- Bootstrap CDN -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js"></script>
</head>
<body>

<c:import url="/WEB-INF/views/include/top_menu.jsp"/>
```

```
<div class="container" style="margin-top:100px">
<div class="row">
	<div class="col-sm-3"></div>
	<div class="col-sm-6">
		<div class="card shadow">
			<div class="card-body">

	<form:form action='${root }board/write_pro' method='post' modelAttribute="writeContentBean">
		<div class="form-group">
			<form:label path="content_subject">제목</form:label>
			<form:input path="content_subject" class='form-control'/>
			<form:errors path='content_subject' style='color:red' />
		</div>
		<div class="form-group">
			<form:label path="content_text">내용</form:label>
			<form:textarea path="content_text" class="form-control" rows="10" style="resize:none"/>
			<form:errors path='content_text' style='color:red'/>
		</div>
		<div class="form-group">
			~~<form:label path="content_file">첨부 이미지</form:label>~~
			~~<form:input type='file' path='content_file' class="form-control" accept="image/*"/>~~
		</div>
		<div class="form-group">
			<div class="text-right">
				<form:button class='btn btn-primary'>작성하기</form:button>
			</div>
		</div>

	</form:form>

				</div>
			</div>
		</div>
		<div class="col-sm-3"></div>
	</div>
</div>

<c:import url="/WEB-INF/views/include/bottom_info.jsp"/>

</body>
</html>
```

4) BoardController.java

```
@GetMapping("/write")
public String write(@ModelAttribute("writeContentBean") ContentBean writeContentBean) {
	return "board/write";
}
```

5) ContentBean.java -유효성 검사를 위한 어노테이션

```
public class ContentBean {
	private int content_idx;

	@NotBlank
	private String content_subject;

	@NotBlank
	private String content_text;

	private String content_file;
	private int content_writer_idx;
	private int content_board_idx;
	private String content_date;
//이하생략
```

6) BoardController.java

```
	@PostMapping("/write_pro")
	public String write_pro(@Valid @ModelAttribute("writeContentBean") ContentBean writeContentBean,
BindingResult result) {
		if(result.hasErrors()) {
			return "board/write";
```

```
                }

                return "board/write_success";
        }
```

7) error_message.properties

```
NotBlank.writeContentBean.content_subject = 제목을 입력해주세요
NotBlank.writeContentBean.content_text = 내용을 입력해주세요
```

(2) 파일 업로드

1) write.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
<c:set var="root" value="${pageContext.request.contextPath }/"/>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>웹 프로젝트</title>
<!-- Bootstrap CDN -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js"></script>
</head>
<body>

<c:import url="/WEB-INF/views/include/top_menu.jsp"/>

<div class="container" style="margin-top:100px">
<div class="row">
        <div class="col-sm-3"></div>
        <div class="col-sm-6">
                <div class="card shadow">
                        <div class="card-body">
        <form:form action='${root }board/write_pro' method='post' modelAttribute="writeContentBean"
enctype="multipart/form-data">
        <div class="form-group">
                <form:label path="content_subject">제목</form:label>
                <form:input path="content_subject" class='form-control'/>
                <form:errors path='content_subject' style='color:red' />
        </div>
        <div class="form-group">
                <form:label path="content_text">내용</form:label>
                <form:textarea path="content_text" class="form-control" rows="10" style="resize:none"/>
                <form:errors path='content_text' style='color:red'/>
        </div>
        <div class="form-group">

                <form:label path="upload_file">첨부 이미지</form:label>

                <form:input type='file' path='upload_file' class="form-control" accept="image/*"/>

        </div>
        <div class="form-group">
                <div class="text-right">
                        <form:button class='btn btn-primary'>작성하기</form:button>
                </div>
        </div>

</form:form>
                                </div>
                        </div>
                </div>
                <div class="col-sm-3"></div>
        </div>
</div>

<c:import url="/WEB-INF/views/include/bottom_info.jsp"/>

</body>
</html>
```

2) ContentBean.java

```
public class ContentBean {
        private int content_idx;

        @NotBlank
        private String content_subject;

        @NotBlank
        private String content_text;

        private MultipartFile upload_file;  //write.jsp에서 전달되는 멀티파트파일

        private String content_file;         //db에서 가져온 파일명을 저장

        private int content_writer_idx;
        private int content_board_idx;
```

3) SpringConfigClass.java

```
protected void customizeRegistration(Dynamic registration) {

        super.customizeRegistration(registration);

        MultipartConfigElement config1 = new MultipartConfigElement(null, 52428800, 524288000, 0); //50MG
        registration.setMultipartConfig(config1);
}
```

4) ServletAppContext.java

```
        @Bean
        public StandardServletMultipartResolver multipartResolver() {   //자동으로 사용됨
                return new StandardServletMultipartResolver();
        }
```

5) BoardService.java

```
@Service
public class BoardService {

        public void addContentInfo(ContentBean writeContentBean) {

                //System.out.println(writeContentBean.getUpload_file());
                //System.out.println(writeContentBean.getUpload_file().getSize());

                MultipartFile upload_file = writeContentBean.getUpload_file();


                if(upload_file.getSize() > 0) {
                        String file_name = saveUploadFile(upload_file);
                        writeContentBean.setContent_file(file_name);
                }

                writeContentBean.setContent_writer_idx(loginUserBean.getUser_idx());

                boardDao.addContentInfo(writeContentBean);
        }
```

6) BoardController.java

```
        @Autowired
        private BoardService boardService;
        @PostMapping("/write_pro")
        public String write_pro(@Valid @ModelAttribute("writeContentBean") ContentBean writeContentBean, 
BindingResult result) {
                if(result.hasErrors()) {
                        return "board/write";
                }

                boardService.addContentInfo(writeContentBean);

                return "board/write_success";
        }
```

7) resources 폴더- upload 폴더 생성

실제 업로드되는 위치 확인해서 수정할 것

```
@Controller
public class HomeController {

        @RequestMapping(value="/", method=RequestMethod.GET)
        public String home(HttpServletRequest request) {

//              System.out.println(request.getServletContext().getRealPath("/"));  //실제경로확인

                return "redirect:main";
        }
```

C:\GitHub\choimysil\spring\.metadata\.plugins\org.eclipse.wst.server.core\tmp2\wtpwebapps\webProjectJava10_6\

8) option.properties -경로 저장

```
path.upload = 
C:/GitHub/choimysil/spring/.metadata/.plugins/org.eclipse.wst.server.core/tmp2/wtpwebapps/webProjectJava1
0_6/resources/upload
```

9) BoardService.java

```
@Service
@PropertySource("/WEB-INF/properties/option.properties")
public class BoardService {

        @Value("${path.upload}")
        private String path_upload;

        @Autowired
        private BoardDao boardDao;

        @Resource(name = "loginUserBean")
        private UserBean loginUserBean;

        private String saveUploadFile(MultipartFile upload_file) {

                String file_name = System.currentTimeMillis() + "_" + upload_file.getOriginalFilename();

                try {
                        upload_file.transferTo(new File(path_upload + "/" + file_name));
                }catch(Exception e) {
                        e.printStackTrace();
                }

                return file_name;
        }

        public void addContentInfo(ContentBean writeContentBean) {

                System.out.println(writeContentBean.getUpload_file());
                System.out.println(writeContentBean.getUpload_file().getSize());

                MultipartFile upload_file = writeContentBean.getUpload_file();
                if(upload_file.getSize() > 0) {
                        String file_name = saveUploadFile(upload_file);
                        writeContentBean.setContent_file(file_name);
                }

                writeContentBean.setContent_writer_idx(loginUserBean.getUser_idx());

                boardDao.addContentInfo(writeContentBean);
        }
}
```

10) BoardController.java - 글쓰기 할때 board_info_idx를 hidden으로 전달할 것

```
        @GetMapping("/write")
        public String write(@ModelAttribute("writeContentBean") ContentBean writeContentBean,
                                                        @RequestParam("board_info_idx") int board_info_idx) {

                writeContentBean.setContent_board_idx(board_info_idx);   //전달된 board_info_idx를 
ContentBean에 저장해둠

                return "board/write";
        }
```

11) write.jsp - content_board_idx를 hidden으로 전달

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
<c:set var="root" value="${pageContext.request.contextPath }/"/>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>웹 프로젝트</title>
<!-- Bootstrap CDN -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js"></script>
</head>
<body>

<c:import url="/WEB-INF/views/include/top_menu.jsp"/>

<div class="container" style="margin-top:100px">
<div class="row">
        <div class="col-sm-3"></div>
        <div class="col-sm-6">
                <div class="card shadow">
                        <div class="card-body">
        <form:form action="${root }board/write_pro' method='post' modelAttribute="writeContentBean"
enctype="multipart/form-data">

           <form:hidden path="content_board_idx"/>

        <div class="form-group">
                <form:label path="content_subject">제목</form:label>
                <form:input path="content_subject" class='form-control'/>
                <form:errors path='content_subject' style='color:red' />
        </div>
        <div class="form-group">
                <form:label path="content_text">내용</form:label>
                <form:textarea path="content_text" class="form-control" rows="10" style="resize:none"/>
                <form:errors path='content_text' style='color:red'/>
        </div>
        <div class="form-group">

                <form:label path="upload_file">첨부 이미지</form:label>

                <form:input type='file' path='upload_file' class="form-control" accept="image/*"/>

        </div>
```

```
		<div class="form-group">
			<div class="text-right">
				<form:button class='btn btn-primary'>작성하기</form:button>
			</div>
		</div>

</form:form>
						</div>
					</div>
				</div>
				<div class="col-sm-3"></div>
		</div>
</div>

<c:import url="/WEB-INF/views/include/bottom_info.jsp"/>

</body>
</html>
```

(3) DB에 저장

1) BoardMapper 인터페이스

파일업로드할때 파일첨부하지 않을 경우 에러가 생긴다. jdbcType=VARCHAR로 설정시 null로 저장되면서 에러나지 않음

```
public interface BoardMapper {

	@Insert("insert into content_table(content_idx, content_subject, content_text, " +
			"content_file, content_writer_idx, content_board_idx, content_date) " +
			"values (content_seq.nextval, #{content_subject}, #{content_text}, #{content_file,
jdbcType=VARCHAR}, " +
			"#{content_writer_idx}, #{content_board_idx}, sysdate)")
	void addContentInfo(ContentBean writeContentBean);
}
```

2) BoardDao.java

```
@Repository
public class BoardDao {

	@Autowired
	private BoardMapper boardMapper;

	public void addContentInfo(ContentBean writeContentBean) {
		boardMapper.addContentInfo(writeContentBean);
	}
}
```

3) BoardService

```
@Service
@PropertySource("/WEB-INF/properties/option.properties")
public class BoardService {

	@Value("${path.upload}")
	private String path_upload;

	@Autowired
	private BoardDao boardDao;

	@Resource(name = "loginUserBean")
	private UserBean loginUserBean;  //로그인사용자의 정보 받아서

	private String saveUploadFile(MultipartFile upload_file) {

		String file_name = System.currentTimeMillis() + "_" + upload_file.getOriginalFilename();

		try {
			upload_file.transferTo(new File(path_upload + "/" + file_name));
		}catch(Exception e) {
			e.printStackTrace();
```

```
                }

                return file_name;
        }

        public void addContentInfo(ContentBean writeContentBean) {

                System.out.println(writeContentBean.getUpload_file());
                System.out.println(writeContentBean.getUpload_file().getSize());

                MultipartFile upload_file = writeContentBean.getUpload_file();

                if(upload_file.getSize() > 0) {
                        String file_name = saveUploadFile(upload_file);
                        writeContentBean.setContent_file(file_name);
                }
                //로그인사용자의 idx를 (contentBean의 writer_idx)에 저장
                writeContentBean.setContent_writer_idx(loginUserBean.getUser_idx());


                boardDao.addContentInfo(writeContentBean);
        }
}
```

4) BoardController.java

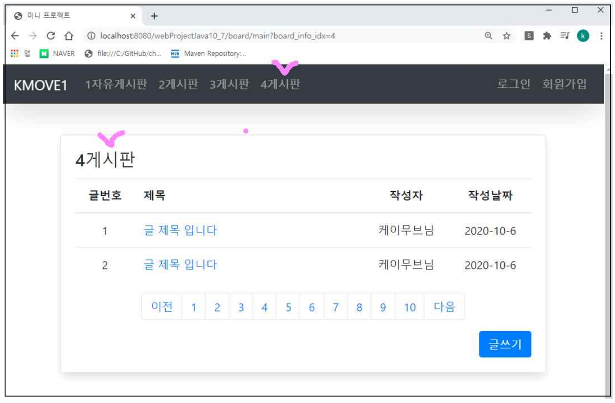
```
@Controller
@RequestMapping("/board")
public class BoardController {

        @Autowired
        private BoardService boardService;

        @PostMapping("/write_pro")
        public String write_pro(@Valid @ModelAttribute("writeContentBean") ContentBean writeContentBean,
BindingResult result) {
                if(result.hasErrors()) {
                        return "board/write";
                }

                boardService.addContentInfo(writeContentBean);

                return "board/write_success";
        }
```

5) write_success.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:set var='root' value='${pageContext.request.contextPath }/'/>
<script>
        alert('저장되었습니다')
        location.href = '${root}board/read'
</script>
```

7. 글 목록 보기

(1) 해당 게시판의 목록보기



4게시판

글번호	제목	작성자	작성날짜
1	글 제목 입니다	케이무브님	2020-10-6
2	글 제목 입니다	케이무브님	2020-10-6

1) BoardMapper -게시판의 board_info_idx 를 전달하여 “게시판 이름”을 조회하는 select

```
	@Select("select board_info_name " +
			"from board_info_table " +
			"where board_info_idx = #{board_info_idx}")
	String getBoardInfoName(int board_info_idx);
```

2) BoardDao.java -dao 작업

```
	public String getBoardInfoName(int board_info_idx) {
		return boardMapper.getBoardInfoName(board_info_idx);
	}
```

3) BoardService -service 작업

```
	public String getBoardInfoName(int board_info_idx) {
		return boardDao.getBoardInfoName(board_info_idx);
	}
```

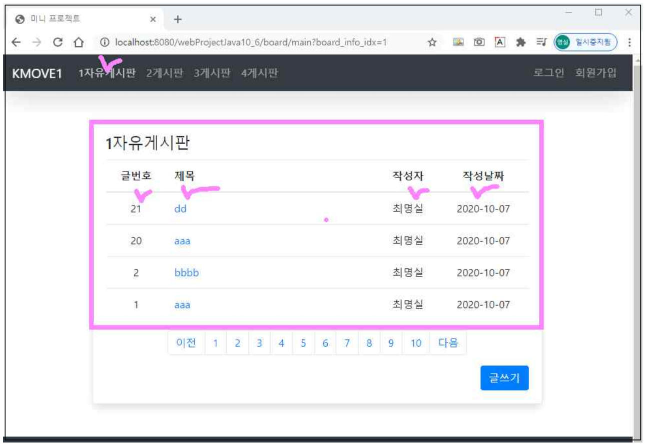
4) BoardController.java

```
	@GetMapping("/main")
	public String main(@RequestParam("board_info_idx") int board_info_idx,
	                                                 Model model) {

		model.addAttribute("board_info_idx", board_info_idx);

		String boardInfoName = boardService.getBoardInfoName(board_info_idx);
		model.addAttribute("boardInfoName", boardInfoName);

		return "board/main";
	}
```

5) board/main.jsp

```
<!-- 게시글 리스트 -->
<div class="container" style="margin-top:100px">
        <div class="card shadow">
                <div class="card-body">
                        <h4 class="card-title">${boardInfoName }</h4>
                        <table class="table table-hover" id='board_list'>
                                <thead>
                                        <tr>
                                                <th class="text-center d-none
d-md-table-cell">글번호</th>
                                                <th class="w-50">제목</th>
                                                <th class="text-center d-none
d-md-table-cell">작성자</th>
                                                <th class="text-center d-none
d-md-table-cell">작성날짜</th>
                                        </tr>
                                </thead>
```

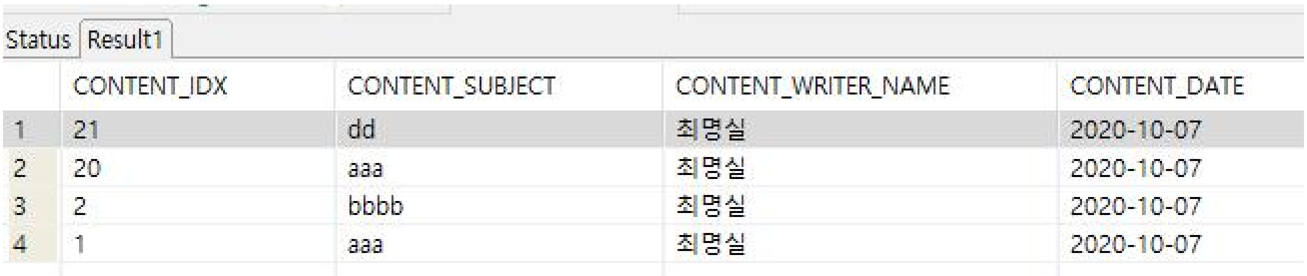
(2) 게시판의 글과 사용자 조인



1자유게시판

글번호	제목	작성자	작성날짜
21	dd	최명실	2020-10-07
20	aaa	최명실	2020-10-07
2	bbbb	최명실	2020-10-07
1	aaa	최명실	2020-10-07

```
select a1.content_idx, a1.content_subject, a2.user_name as content_writer_name,
   to_char(a1.content_date, 'YYYY-MM-DD') as content_date
   from content_table a1, user_table a2
   where a1.content_writer_idx = a2.user_idx  and a1.content_board_idx = 1
   order by a1.content_idx desc;
```



	CONTENT_IDX	CONTENT_SUBJECT	CONTENT_WRITER_NAME	CONTENT_DATE
1	21	dd	최명실	2020-10-07
2	20	aaa	최명실	2020-10-07
3	2	bbbb	최명실	2020-10-07
4	1	aaa	최명실	2020-10-07

1) BoardMapper.java

```
        @Select("select a1.content_idx, a1.content_subject, a2.user_name as content_writer_name, " +
                        "        to_char(a1.content_date, 'YYYY-MM-DD') as content_date " +
                        "from content_table a1, user_table a2 " +
                        "where a1.content_writer_idx = a2.user_idx " +
                        "       and a1.content_board_idx = #{board_info_idx} " +
                        "order by a1.content_idx desc")
        List<ContentBean> getContentList(int board_info_idx);
```

2) ContentBean.java - 조인에 의해 조회된 결과를 저장할 필드 추가

```
public class ContentBean {
        private int content_idx;

        @NotBlank
        private String content_subject;

        @NotBlank
        private String content_text;

        private MultipartFile upload_file;   //write.jsp에서 전달되는 멀티파트파일

        private String content_file;         //db에서 가져온 파일명을 저장

        private int content_writer_idx;
        private int content_board_idx;

        private String content_date;          //날짜
        private String content_writer_name;   //작성자   조인에 의해 select된 결과를 저장할 필드 추가
```

3) BoardDao.java

```
        public List<ContentBean> getContentList(int board_info_idx){
                return boardMapper.getContentList(board_info_idx);
        }
```

4) BoardService.java

```
        public List<ContentBean> getContentList(int board_info_idx){
                return boardDao.getContentList(board_info_idx);
        }
```

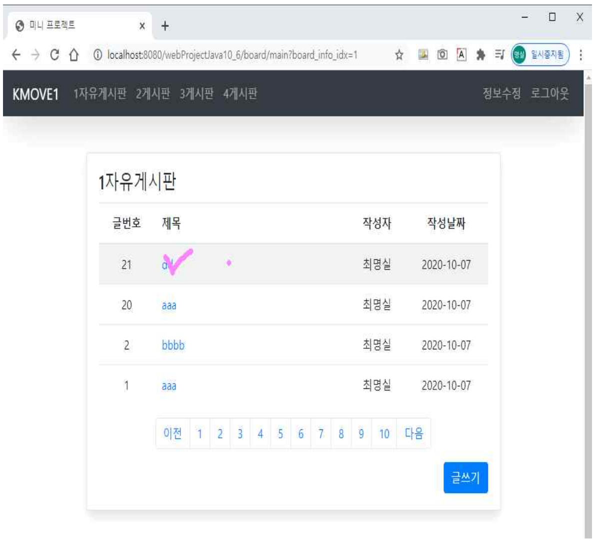
5) BoardController.java

```
        @GetMapping("/main")
        public String main(@RequestParam("board_info_idx") int board_info_idx,
                                                   Model model) {

                model.addAttribute("board_info_idx", board_info_idx);
                String boardInfoName = boardService.getBoardInfoName(board_info_idx);
                model.addAttribute("boardInfoName", boardInfoName);

                List<ContentBean> contentList = boardService.getContentList(board_info_idx);
                model.addAttribute("contentList", contentList);

                return "board/main";
        }
```

6) board/main.jsp

```
<table class="table table-hover" id='board_list'>
        <thead>
                <tr>
                        <th class="text-center d-none d-md-table-cell">글번호</th>
                        <th class="w-50">제목</th>
                        <th class="text-center d-none d-md-table-cell">작성자</th>
                        <th class="text-center d-none d-md-table-cell">작성날짜</th>
                </tr>
        </thead>

        <tbody>
                <c:forEach var='obj' items="${contentList }">
                <tr>
                        <td class="text-center d-none d-md-table-cell">${obj.content_idx }</td>
                        <td><a href='${root }board/read'>${obj.content_subject }</a></td>
                        <td class="text-center d-none d-md-table-cell">${obj.content_writer_name }</td>
                        <td class="text-center d-none d-md-table-cell">${obj.content_date }</td>

                </tr>
                </c:forEach>
        </tbody>
</table>
```

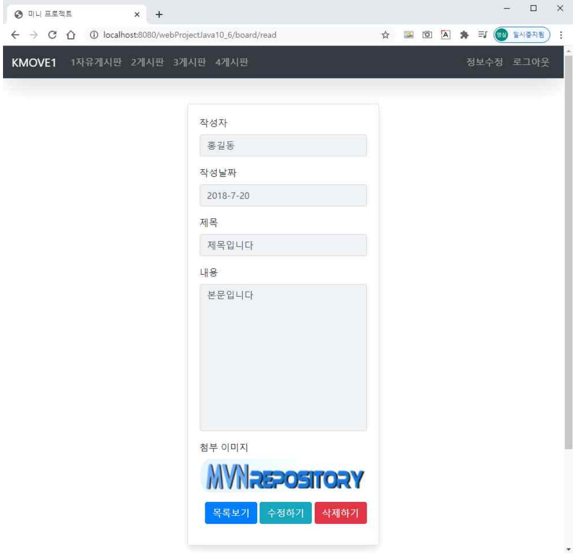
(2) 글제목을 링크시 --> 해당 글번호의 내용을 read.jsp에 읽어오기



미니 프로젝트
localhost:8080/webProjectJava10_6/board/main?board_info_idx=1
KMOVE1 1자유게시판 2게시판 3게시판 4게시판 정보수정 로그아웃

1자유게시판

글번호	제목	작성자	작성날짜
21	a	최명실	2020-10-07
20	aaa	최명실	2020-10-07
2	bbbb	최명실	2020-10-07
1	aaa	최명실	2020-10-07



미니 프로젝트
localhost:8080/webProjectJava10_6/board/read
KMOVE1 1자유게시판 2게시판 3게시판 4게시판 정보수정 로그아웃

작성자: 홍길동
작성날짜: 2018-7-20
제목: 제목입니다
내용: 본문입니다
첨부 이미지

1) board/main.jsp

```
        <tbody>
                <c:forEach var='obj' items="${contentList }">
                <tr>
                        <td class="text-center d-none d-md-table-cell">${obj.content_idx }</td>
<td><a href="${root }board/read?board_info_idx=${board_info_idx}&content_idx=${obj.content_idx} ">${obj.content_subject
}</a></td>
                        <td class="text-center d-none d-md-table-cell">${obj.content_writer_name }</td>
                        <td class="text-center d-none d-md-table-cell">${obj.content_date }</td>

                </tr>
                </c:forEach>
        </tbody>
```

2) BoardController.java -jsp에서 링크시 전달한 파라메타 전달받기

```
        @GetMapping("/read")
        public String read(@RequestParam("board_info_idx") int board_info_idx,
                                            @RequestParam("content_idx") int content_idx,
                                            Model model) {

                model.addAttribute("board_info_idx", board_info_idx);

                return "board/read";
        }
```

3) BoardMapper.java

```
        @Select("select a2.user_name as content_writer_name, " +
                        "       to_char(a1.content_date, 'YYYY-MM-DD') as content_date, " +
                        "       a1.content_subject, a1.content_text, a1.content_file " +
                        "from content_table a1, user_table a2 " +
                        "where a1.content_writer_idx = a2.user_idx " +
                        "      and content_idx = #{content_idx}")
        ContentBean getContentInfo(int content_idx);
```

4) BoardDao.java

```
	public ContentBean getContentInfo(int content_idx) {
		return boardMapper.getContentInfo(content_idx);
	}
```

5) BoardService.java

```
	public ContentBean getContentInfo(int content_idx) {
		return boardDao.getContentInfo(content_idx);
	}
```

6) BoardController.java

```
@GetMapping("/read")
public String read(@RequestParam("board_info_idx") int board_info_idx,
                                        @RequestParam("content_idx") int content_idx,
                                        Model model) {

	model.addAttribute("board_info_idx", board_info_idx);

	ContentBean readContentBean = boardService.getContentInfo(content_idx);
	model.addAttribute("readContentBean", readContentBean);

	return "board/read";
}
```

7) board/read.jsp

```
<body>

<c:import url="/WEB-INF/views/include/top_menu.jsp" />

<div class="container" style="margin-top:100px">
<div class="row">
	<div class="col-sm-3"></div>
	<div class="col-sm-6">
		<div class="card shadow">
			<div class="card-body">
				<div class="form-group">
					<label for="board_writer_name">작성자</label>
					<input type="text" id="board_writer_name"
name="board_writer_name" class="form-control" value="${readContentBean.content_writer_name }"
disabled="disabled"/>
	</div>
	<div class="form-group">
		<label for="board_date">작성날짜</label>
		<input type="text" id="board_date" name="board_date" class="form-control"
value="${readContentBean.content_date }" disabled="disabled"/>
	</div>
	<div class="form-group">
		<label for="board_subject">제목</label>
		<input type="text" id="board_subject" name="board_subject" class="form-control"
value="${readContentBean.content_subject }" disabled="disabled"/>
	</div>
	<div class="form-group">
		<label for="board_content">내용</label>
		<textarea id="board_content" name="board_content" class="form-control" rows="10"
style="resize:none" disabled="disabled">${readContentBean.content_text }</textarea>
	</div>
	<c:if test="${readContentBean.content_file != null }">
	<div class="form-group">
		<label for="board_file">첨부 이미지</label>
		<img src="${root }upload/${readContentBean.content_file}" width="100%"/>

	</div>
	</c:if>
	<div class="form-group">
		<div class="text-right">
			<a href="${root }board/main?board_info_idx=${board_info_idx}" class="btn
btn-primary">목록보기</a>
			<a href="${root }board/modify" class="btn btn-info">수정하기</a>
```

```
            <a href="${root }board/delete" class="btn btn-danger">삭제하기</a>
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <div class="col-sm-3"></div>
    </div>
</div>

<c:import url="/WEB-INF/views/include/bottom_info.jsp" />

</body>
```

(3) 글쓰기한 후 <u>content_idx가 증가</u>된 상태의 값을 전달하도록 변경

1) BoardMapper.java

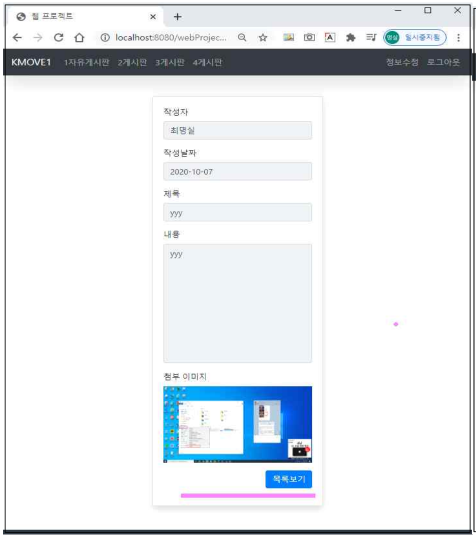
```
@SelectKey(statement = "select content_seq.nextval from dual", keyProperty = "content_idx", before = true, resultType = 
int.class)

@Insert("insert into content_table(content_idx, content_subject, content_text, " +
                              "content_file, content_writer_idx, content_board_idx, content_date) " +
                              "values (#{content_idx}, #{content_subject}, #{content_text}, #{content_file, 
jdbcType=VARCHAR}, " +
                              "#{content_writer_idx}, #{content_board_idx}, sysdate)")
        void addContentInfo(ContentBean writeContentBean);
```

2) write_success.jsp

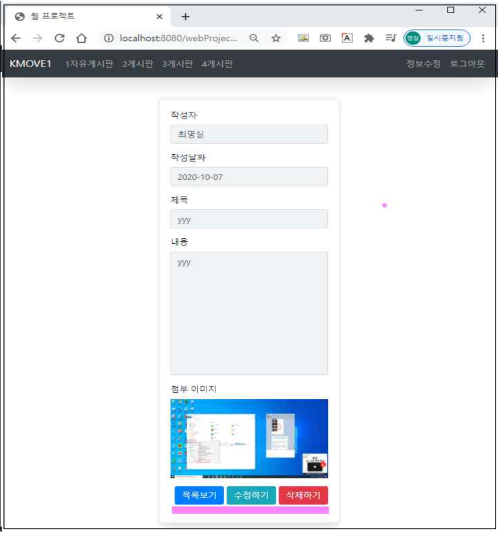
```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:set var='root' value='${pageContext.request.contextPath }/'/>
<script>
        alert('저장되었습니다')
        location.href = 
'${root}board/read?board_info_idx=${writeContentBean.content_board_idx}&content_idx=${writeContentBean.co
ntent_idx}'
</script>
```

8. 글 수정하기

(1) 로그인한 사용자만 수정하기,삭제하기 버튼 동작하기



작성자
최명실
작성날짜
2020-10-07
제목
yyy
내용
yyy
첨부 이미지



작성자
최명실
작성날짜
2020-10-07
제목
yyy
내용
yyy
첨부 이미지

1) BoardController.java

```
        @GetMapping("/read")
        public String read(@RequestParam("board_info_idx") int board_info_idx,
                                                  @RequestParam("content_idx") int content_idx,
                                                  Model model) {

                model.addAttribute("board_info_idx", board_info_idx);

                model.addAttribute("content_idx", content_idx);    //글의 idx전달하기 위함

                ContentBean readContentBean = boardService.getContentInfo(content_idx);
                model.addAttribute("readContentBean", readContentBean);

                return "board/read";
        }
```

2) BoardMapper.java

```
        @Select("select a2.user_name as content_writer_name, " +
                        "       to_char(a1.content_date, 'YYYY-MM-DD') as content_date, " +
                        "       a1.content_subject, a1.content_text, a1.content_file, a1.content_writer_idx "
+
                        "from content_table a1, user_table a2 " +
                        "where a1.content_writer_idx = a2.user_idx " +
                        "      and content_idx = #{content_idx}")
        ContentBean getContentInfo(int content_idx);
```

3) BoardController.java

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

        @Autowired
        private BoardService boardService;

        //로그인한 사람의 정보 받아오기
        @Resource(name = "loginUserBean")
        private UserBean loginUserBean;

        @GetMapping("/read")
        public String read(@RequestParam("board_info_idx") int board_info_idx,
                                                  @RequestParam("content_idx") int content_idx,
                                                  Model model) {

                model.addAttribute("board_info_idx", board_info_idx);
                model.addAttribute("content_idx", content_idx);    //글의 idx전달하기 위함

                ContentBean readContentBean = boardService.getContentInfo(content_idx);
                model.addAttribute("readContentBean", readContentBean);

                model.addAttribute("loginUserBean", loginUserBean);

                return "board/read";
        }
```

4) read.jsp

```
<body>

<c:import url="/WEB-INF/views/include/top_menu.jsp" />

<div class="container" style="margin-top:100px">
<div class="row">
        <div class="col-sm-3"></div>
        <div class="col-sm-6">
                <div class="card shadow">
                        <div class="card-body">
                                <div class="form-group">
```

```
                                        <label for="board_writer_name">작성자</label>
                                        <input type="text" id="board_writer_name"
name="board_writer_name" class="form-control" value="${readContentBean.content_writer_name }"
disabled="disabled"/>
        </div>
        <div class="form-group">
                <label for="board_date">작성날짜</label>
                <input type="text" id="board_date" name="board_date" class="form-control"
value="${readContentBean.content_date }" disabled="disabled"/>
        </div>
        <div class="form-group">
                <label for="board_subject">제목</label>
                <input type="text" id="board_subject" name="board_subject" class="form-control"
value="${readContentBean.content_subject }" disabled="disabled"/>
        </div>
        <div class="form-group">
                <label for="board_content">내용</label>
                <textarea id="board_content" name="board_content" class="form-control" rows="10"
style="resize:none" disabled="disabled">${readContentBean.content_text }</textarea>
        </div>
        <c:if test="${readContentBean.content_file != null }">
        <div class="form-group">
                <label for="board_file">첨부 이미지</label>
                <img src="${root }upload/${readContentBean.content_file}" width="100%"/>

        </div>
    </c:if>
        <div class="form-group">
                <div class="text-right">
                <a href="${root }board/main?board_info_idx=${board_info_idx}" class="btn
btn-primary">목록보기</a>
                <c:if test="${loginUserBean.user_idx == readContentBean.content_writer_idx }">
                <a href="${root }board/modify?board_info_idx=${board_info_idx}&content_idx=${content_idx}"
class="btn btn-info">수정하기</a>
                <a href="${root }board/delete?board_info_idx=${board_info_idx}&content_idx=${content_idx}"
class="btn btn-danger">삭제하기</a>
                </c:if>
                </div>
        </div>
        </div>
        </div>
        </div>
        <div class="col-sm-3"></div>
        </div>
</div>


<c:import url="/WEB-INF/views/include/bottom_info.jsp"/>

</body>
</html>
```

(2) 로그인한 사용자만 URL(주소)를 통하여 접근가능하도록 인터셉터 지정

1) CheckWriterInterceptor.java -인터셉터 만들기

```
public class CheckWriterInterceptor implements HandlerInterceptor{

        private UserBean loginUserBean;
        private BoardService boardService;

        public CheckWriterInterceptor(UserBean loginUserBean, BoardService boardService) {
                // TODO Auto-generated constructor stub
                this.loginUserBean = loginUserBean;
                this.boardService = boardService;
        }

        @Override
        public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object
handler)
                        throws Exception {
                // TODO Auto-generated method stub
```

```
		String str1 = request.getParameter("content_idx");
		int content_idx = Integer.parseInt(str1);

		ContentBean currentContentBean = boardService.getContentInfo(content_idx);

		if(currentContentBean.getContent_writer_idx() != loginUserBean.getUser_idx()) {
			String contextPath = request.getContextPath();
			response.sendRedirect(contextPath + "/board/not_writer");
			return false;
		}

		return true;
	}
}
```

2) BoardController.java

```
	@GetMapping("/not_writer")
	public String not_writer() {
		return "board/not_writer";
	}
```

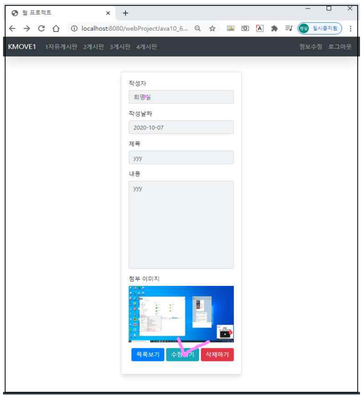
3) board/not_writer.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:set var='root' value='${pageContext.request.contextPath }/'/>
<script>
	alert('잘못된 접근입니다')
	location.href = '${root}main'
</script>
```

4) ServletAppContext.java 로그인한 사용자만 url에서 접근 modify,delete가능

```
	@Autowired
	private BoardService boardService;


@Override
	public void addInterceptors(InterceptorRegistry registry) {
		// TODO Auto-generated method stub
		WebMvcConfigurer.super.addInterceptors(registry);

		TopMenuInterceptor topMenuInterceptor = new TopMenuInterceptor(topMenuService,
loginUserBean);

		InterceptorRegistration reg1 = registry.addInterceptor(topMenuInterceptor);
		reg1.addPathPatterns("/**");

		CheckLoginInterceptor checkLoginInterceptor = new CheckLoginInterceptor(loginUserBean);
		InterceptorRegistration reg2 = registry.addInterceptor(checkLoginInterceptor);
		reg2.addPathPatterns("/user/modify", "/user/logout", "/board/*");
		reg2.excludePathPatterns("/board/main");

		CheckWriterInterceptor checkWriterInterceptor = new
CheckWriterInterceptor(loginUserBean, boardService);
		InterceptorRegistration reg3 = registry.addInterceptor(checkWriterInterceptor);
		reg3.addPathPatterns("/board/modify", "/board/delete");
	}
```

(3) 글 수정하기



작성자
최명실

작성날짜
2020-10-07

제목
yyy

내용
yyy

첨부 이미지

1) BoardController.java

```
        @GetMapping("/modify")
        public String modify(@RequestParam("board_info_idx") int board_info_idx,
                                                        @RequestParam("content_idx") int content_idx,
                                                        @ModelAttribute("modifyContentBean") ContentBean
modifyContentBean, Model model) {

                model.addAttribute("board_info_idx", board_info_idx);
                model.addAttribute("content_idx", content_idx);

                ContentBean tempContentBean = boardService.getContentInfo(content_idx);

                modifyContentBean.setContent_writer_name(tempContentBean.getContent_writer_name());
                modifyContentBean.setContent_date(tempContentBean.getContent_date());
                modifyContentBean.setContent_subject(tempContentBean.getContent_subject());
                modifyContentBean.setContent_text(tempContentBean.getContent_text());
                modifyContentBean.setContent_file(tempContentBean.getContent_file());
                modifyContentBean.setContent_writer_idx(tempContentBean.getContent_writer_idx());
                modifyContentBean.setContent_board_idx(board_info_idx);
                modifyContentBean.setContent_idx(content_idx);

                return "board/modify";
        }
```

2) modify.jsp

```
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>


<div class="card-body">
        <form:form action= ${root }board/modify_pro' method='post' modelAttribute="modifyContentBean"
enctype="multipart/form-data">
                <form:hidden path="content_idx"/>
                <form:hidden path="content_board_idx"/>
                <div class="form-group">
                        <form:label path="content_writer_name">작성자</form:label>
                        <form:input path="content_writer_name" class='form-control' readonly="true"/>
                </div>
                <div class="form-group">
                        <form:label path="content_date">작성날짜</form:label>
                        <form:input path="content_date" class='form-control' readonly='true'/>
                </div>
                <div class="form-group">
                        <form:label path="content_subject">제목</form:label>
                        <form:input path="content_subject" class='form-control'/>
                        <form:errors path="content_subject" style='color:red'/>
                </div>
                <div class="form-group">
                        <form:label path="content_text">내용</form:label>
                        <form:textarea path="content_text" class="form-control" rows="10"
style="resize:none"/>
                        <form:errors path="content_text" style='color:red'/>
                </div>
                <div class="form-group">
                        <label for="board_file">첨부 이미지</label>
                        <c:if test="${modifyContentBean.content_file != null }">
                        <img src="${root }upload/${modifyContentBean.content_file}" width="100%"/>
                        <form:hidden path="content_file"/>
                        </c:if>
                        <form:input path="upload_file" type='file' class="form-control" accept="image/*"/>
                </div>
                <div class="form-group">
                        <div class="text-right">
                                <form:button class='btn btn-primary'>수정완료</form:button>
                                <a href="${root
}board/read?board_info_idx=${board_info_idx}&content_idx=${content_idx}" class="btn btn-info">취소</a>
                        </div>
                </div>
        </form:form>
</div>
```

(2) 수정하기 버튼시 유효성 검사

1) error_message.properties

```
NotBlank.modifyContentBean.content_subject = 제목을 입력해주세요
NotBlank.modifyContentBean.content_text = 내용을 입력해주세요
```

2) BoardController.java

```
@PostMapping("/modify_pro")
        public String modify_pro(@Valid @ModelAttribute("modifyContentBean") ContentBean
modifyContentBean, BindingResult result) {
                if(result.hasErrors()) {
                        return "board/modify";
                }

                //

                return "board/modify_success";
        }
```

(3) db에 수정하기(update)

1) BoardMapper.java

```
        @Update("update content_table " +
                        "set content_subject = #{content_subject}, content_text = #{content_text}, " +
                        "content_file = #{content_file, jdbcType=VARCHAR} " +
                        "where content_idx = #{content_idx}")
        void modifyContentInfo(ContentBean modifyContentBean);
```

2) BoardDao.java

```
        public void modifyContentInfo(ContentBean modifyContentBean) {
                boardMapper.modifyContentInfo(modifyContentBean);
        }
```

3) BoardService.java

```
        public void modifyContentInfo(ContentBean modifyContentBean) {

                MultipartFile upload_file = modifyContentBean.getUpload_file();

                if(upload_file.getSize() > 0) {
                        String file_name = saveUploadFile(upload_file);
                        modifyContentBean.setContent_file(file_name);
                }

                boardDao.modifyContentInfo(modifyContentBean);
        }
```

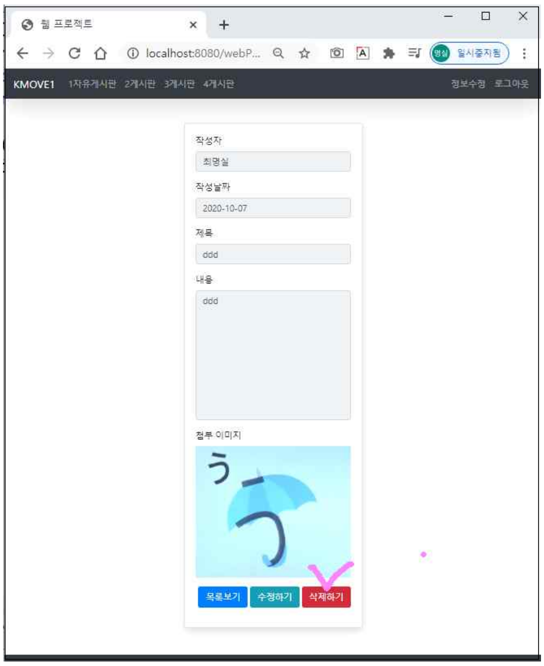
4) BoardController.java

```
	@PostMapping("/modify_pro")
	public String modify_pro(@Valid @ModelAttribute("modifyContentBean") ContentBean
modifyContentBean,
						BindingResult result) {
		if(result.hasErrors()) {
			return "board/modify";
		}
		boardService.modifyContentInfo(modifyContentBean);

		return "board/modify_success";
	}
```

5) modify_success.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:set var='root' value='${pageContext.request.contextPath}/'/>
<script>
	alert('수정되었습니다')
	location.href =
'${root}board/read?board_info_idx=${modifyContentBean.content_board_idx}&content_idx=${modifyContentBean
.content_idx}'
</script>
```

9. 글 삭제하기



작성자

최명실

작성날짜

2020-10-07

제목

ddd

내용

ddd

첨부 이미지

(1) 글 삭제하기

1) BoardMapper.java

```
	@Delete("delete from content_table where content_idx = #{content_idx}")
	void deleteContentInfo(int content_idx);
```

2) BoardDao.java

```
	public void deleteContentInfo(int content_idx) {
		boardMapper.deleteContentInfo(content_idx);
	}
```

3) BoardService.java

```
	public void deleteContentInfo(int content_idx) {
		boardDao.deleteContentInfo(content_idx);
	}
```

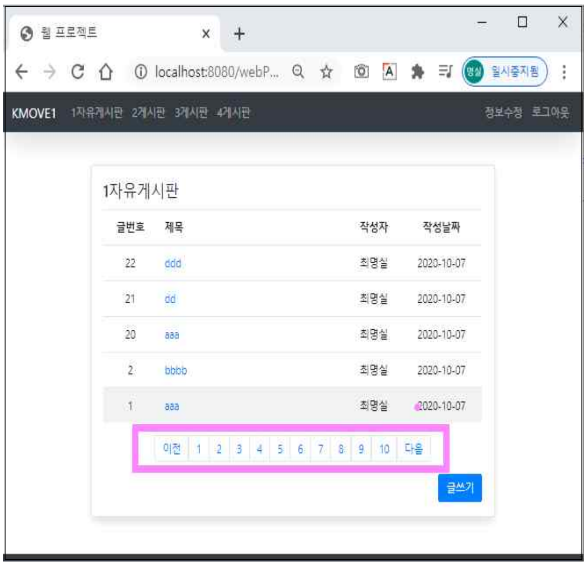
4) BoardController.java

```
	@GetMapping("/delete")
	public String delete(@RequestParam("board_info_idx") int board_info_idx,
					     @RequestParam("content_idx") int content_idx,
					     Model model) {
		
		boardService.deleteContentInfo(content_idx);
		
		model.addAttribute("board_info_idx", board_info_idx);
		
		return "board/delete";
	}
```

5) delete.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:set var="root" value="${pageContext.request.contextPath }/"/>
<script>
	alert('삭제되었습니다')
	location.href='${root}board/main?board_info_idx=${board_info_idx}'
</script>
```

10. 페이징 구현하기



1자유게시판

글번호	제목	작성자	작성날짜
22	ddd	최명실	2020-10-07
21	dd	최명실	2020-10-07
20	aaa	최명실	2020-10-07
2	bbbb	최명실	2020-10-07
1	aaa	최명실	2020-10-07

(1) 한 화면에 10개씩 보이기

1) option.properties

```
page.listcnt = 10
page.paginationcnt = 10
```

2) BoardMapper.java

```
        @Select("select a1.content_idx, a1.content_subject, a2.user_name as content_writer_name, " +
                        "       to_char(a1.content_date, 'YYYY-MM-DD') as content_date " +
                        "from content_table a1, user_table a2 " +
                        "where a1.content_writer_idx = a2.user_idx " +
                        "      and a1.content_board_idx = #{board_info_idx} " +
                        "order by a1.content_idx desc")
        List<ContentBean> getContentList(int board_info_idx, RowBounds rowRounds);
```

3) BoardDao.java

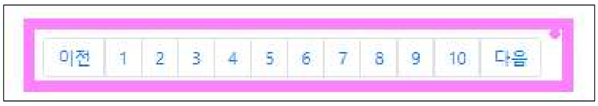
```
        public List<ContentBean> getContentList(int board_info_idx, RowBounds rowBounds){
                return boardMapper.getContentList(board_info_idx, rowBounds);
        }
```

4) BoardService.java

```
        @Value("${page.listcnt}")
        private int page_listcnt;      //개수 10개

        @Value("${page.paginationcnt}")
        private int page_paginationcnt;

public List<ContentBean> getContentList(int board_info_idx, int page){

                int start = (page - 1) * page_listcnt;                //1 -> (1-1)*10=0
                RowBounds rowBounds = new RowBounds(start, page_listcnt);  // 0부터 10개

                return boardDao.getContentList(board_info_idx, rowBounds);
        }
```

http://localhost:8080/webProjectJava10_6/board/main?board_info_idx=2&page=3

(2) 이전, 다음, 페이지 구현



1) PageBean.java

```
package kr.co.korea.beans;

public class PageBean {

        private int min;  // 최소 페이지 번호
        private int max;  // 최대 페이지 번호
        private int prevPage; // 이전 버튼의 페이지 번호
        private int nextPage;// 다음 버튼의 페이지 번호
        private int pageCnt;// 전체 페이지 개수
        private int currentPage; // 현재 페이지 번호

        // contentCnt : 전체글 개수, currentPage : 현재 글 번호, contentPageCnt : 페이지당 글의 개수,
paginationCnt : 페이지 버튼의 개수
        public PageBean(int contentCnt, int currentPage, int contentPageCnt, int paginationCnt) {

                this.currentPage = currentPage; // 현재 페이지 번호

                pageCnt = contentCnt / contentPageCnt; // 전체 페이지 개수
                if(contentCnt % contentPageCnt > 0) {
                        pageCnt++;
                }

                min = ((currentPage - 1) / contentPageCnt) * contentPageCnt + 1;
                max = min + paginationCnt - 1;

                if(max > pageCnt) {
                        max = pageCnt;
                }
                prevPage = min - 1;
                nextPage = max + 1;
                if(nextPage > pageCnt) {
                        nextPage = pageCnt;
                }
        }

        public int getMin() {
                return min;
        }
        public int getMax() {
                return max;
        }
//이하생략
```

2) BoardMapper.java

```
        //글의 갯수
        @Select("select count(*) from content_table where content_board_idx = #{content_board_idx}")
        int getContentCnt(int content_board_idx);
```

3) BoardDao.java

```
        public int getContentCnt(int content_board_idx) {
                return boardMapper.getContentCnt(content_board_idx);
        }
```

4) BoardService.java

```
        @Value("${page.listcnt}")
        private int page_listcnt;       //개수 10개

        @Value("${page.paginationcnt}")
        private int page_paginationcnt;


        public PageBean getContentCnt(int content_board_idx, int currentPage) {

                int content_cnt = boardDao.getContentCnt(content_board_idx);

                PageBean pageBean = new PageBean(content_cnt, currentPage, page_listcnt,
page_paginationcnt);

                return pageBean;
        }
```

5) BoardController.java

```
@GetMapping("/main")
        public String main(@RequestParam("board_info_idx") int board_info_idx,
                                                @RequestParam(value = "page", defaultValue = "1") int page,
                                                Model model) {

                model.addAttribute("board_info_idx", board_info_idx);

                String boardInfoName = boardService.getBoardInfoName(board_info_idx);
                model.addAttribute("boardInfoName", boardInfoName);

                List<ContentBean> contentList = boardService.getContentList(board_info_idx, page);
                model.addAttribute("contentList", contentList);

                PageBean pageBean = boardService.getContentCnt(board_info_idx, page);
                model.addAttribute("pageBean", pageBean);

                return "board/main";
        }
```

6) main.jsp

```
<div class="d-none d-md-block">
        <ul class="pagination justify-content-center">
            <c:choose>
        <c:when test="${pageBean.prevPage <= 0 }">
        <li class="page-item disabled">
            <a href="#" class="page-link">이전</a>
        </li>
        </c:when>
        <c:otherwise>
        <li class="page-item">
            <a href="${root
}board/main?board_info_idx=${board_info_idx}&page=${pageBean.prevPage}" class="page-link">이전</a>
        </li>
        </c:otherwise>
        </c:choose>


        <c:forEach var='idx' begin="${pageBean.min }" end="${pageBean.max }">
        <c:choose>
        <c:when test="${idx == pageBean.currentPage }">
        <li class="page-item active">
            <a href="${root }board/main?board_info_idx=${board_info_idx}&page=${idx}"
class="page-link">${idx }</a>
        </li>
        </c:when>
        <c:otherwise>
        <li class="page-item">
            <a href="${root }board/main?board_info_idx=${board_info_idx}&page=${idx}"
class="page-link">${idx }</a>
        </li>
        </c:otherwise>
        </c:choose>

        </c:forEach>

        <c:choose>
        <c:when test="${pageBean.max >= pageBean.pageCnt }">
        <li class="page-item disabled">
            <a href="#" class="page-link">다음</a>
        </li>
        </c:when>
        <c:otherwise>
        <li class="page-item">
            <a href="${root
}board/main?board_info_idx=${board_info_idx}&page=${pageBean.nextPage}" class="page-link">다음</a>
        </li>
        </c:otherwise>
        </c:choose>

    </ul>
</div>

<div class="text-right">
    <a href="${root }board/write?board_info_idx=${board_info_idx}" class="btn btn-primary">글쓰기</a>
            </div>

        </div>
    </div>
</div>

<c:import url="/WEB-INF/views/include/bottom_info.jsp"/>

</body>
</html>
```

(3) 내용보기 후 목록보기로 갈때 --> 페이지 함께 이동

1) BoardController.java

```
    @GetMapping("/main")
    public String main(@RequestParam("board_info_idx") int board_info_idx,
                                            @RequestParam(value = "page", defaultValue = "1") int page,
                                        Model model) {

                model.addAttribute("board_info_idx", board_info_idx);

                String boardInfoName = boardService.getBoardInfoName(board_info_idx);
                model.addAttribute("boardInfoName", boardInfoName);

                List<ContentBean> contentList = boardService.getContentList(board_info_idx, page);
                model.addAttribute("contentList", contentList);

                PageBean pageBean = boardService.getContentCnt(board_info_idx, page);
                model.addAttribute("pageBean", pageBean);

                model.addAttribute("page", page);

                return "board/main";
        }

        @GetMapping("/read")
        public String read(@RequestParam("board_info_idx") int board_info_idx,
                                        @RequestParam("content_idx") int content_idx,
                                        @RequestParam("page") int page,
                                        Model model) {

                model.addAttribute("board_info_idx", board_info_idx);
                model.addAttribute("content_idx", content_idx);

                ContentBean readContentBean = boardService.getContentInfo(content_idx);
                model.addAttribute("readContentBean", readContentBean);

                model.addAttribute("loginUserBean", loginUserBean);

                model.addAttribute("page", page);

                return "board/read";
        }
```

2) main.jsp

```
<tbody>
        <c:forEach var='obj' items="${contentList }">
        <tr>
                <td class="text-center d-none d-md-table-cell">${obj.content_idx }</td>
                <td><a href="${root
}board/read?board_info_idx=${board_info_idx}&content_idx=${obj.content_idx}&page=${page}">${obj.content_sub
ject }</a></td>
                <td class="text-center d-none d-md-table-cell">${obj.content_writer_name }</td>
                <td class="text-center d-none d-md-table-cell">${obj.content_date }</td>

        </tr>
        </c:forEach>
</tbody>
```

3) read.jsp

```
        <div class="form-group">
                <div class="text-right">
        <a href="${root }board/main?board_info_idx=${board_info_idx}&page=${page}" class="btn
btn-primary">목록보기</a>
        <c:if test="${loginUserBean.user_idx == readContentBean.content_writer_idx }">
        <a href="${root
}board/modify?board_info_idx=${board_info_idx}&content_idx=${content_idx}&page=${page}" class="btn
btn-info">수정하기</a>
<a href="${root }board/delete?board_info_idx=${board_info_idx}&content_idx=${content_idx}" class="btn
btn-danger">삭제하기</a>
</c:if>
                </div>
        </div>
```

(4) 수정후 후 목록보기로 갈때 --> 페이지 함께 이동

1) BoardController.java

```
          @GetMapping("/modify")
          public String modify(@RequestParam("board_info_idx") int board_info_idx,
                                                          @RequestParam("content_idx") int content_idx,
                                                          @ModelAttribute("modifyContentBean") ContentBean
modifyContentBean,
                                                          @RequestParam("page") int page,
                                                          Model model) {

                    model.addAttribute("board_info_idx", board_info_idx);
                    model.addAttribute("content_idx", content_idx);

                    model.addAttribute("page", page);

                    ContentBean tempContentBean = boardService.getContentInfo(content_idx);

                    modifyContentBean.setContent_writer_name(tempContentBean.getContent_writer_name());
                    modifyContentBean.setContent_date(tempContentBean.getContent_date());
                    modifyContentBean.setContent_subject(tempContentBean.getContent_subject());
                    modifyContentBean.setContent_text(tempContentBean.getContent_text());
                    modifyContentBean.setContent_file(tempContentBean.getContent_file());
                    modifyContentBean.setContent_writer_idx(tempContentBean.getContent_writer_idx());
                    modifyContentBean.setContent_board_idx(board_info_idx);
                    modifyContentBean.setContent_idx(content_idx);

                    return "board/modify";
          }

          @PostMapping("/modify_pro")
          public String modify_pro(@Valid @ModelAttribute("modifyContentBean") ContentBean
modifyContentBean, BindingResult result,
@RequestParam("page") int page,     Model model) {

                    model.addAttribute("page", page);

                    if(result.hasErrors()) {
                              return "board/modify";
                    }

                    boardService.modifyContentInfo(modifyContentBean);

                    return "board/modify_success";
          }
```

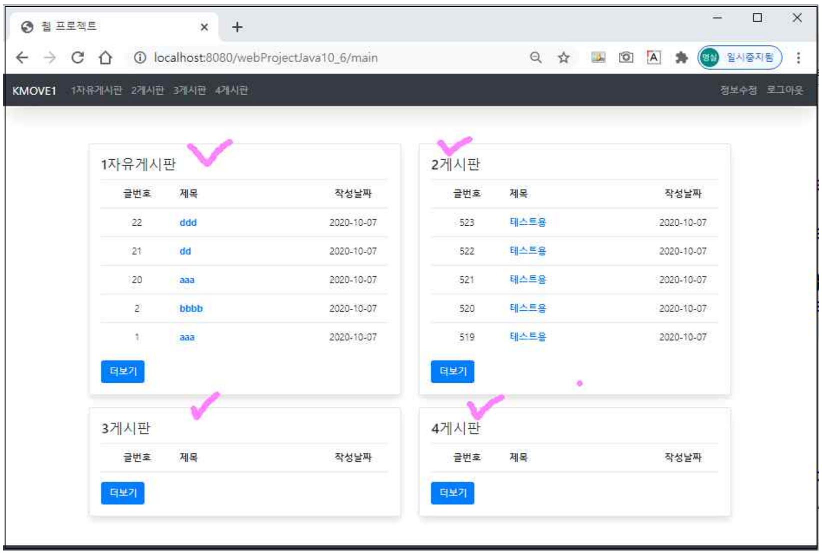
2) modify.jsp

```
<form:form action='${root }board/modify_pro' method='post'
          modelAttribute="modifyContentBean" enctype="multipart/form-data">
<form:hidden path="content_idx"/>
<form:hidden path="content_board_idx"/>
<input type='hidden' name='page' value='${page }'/>

          <div class="form-group">
                    <form:label path="content_writer_name">작성자</form:label>
                    <form:input path="content_writer_name" class='form-control'
                              readonly="true" />
          </div>

 //생략


<div class="form-group">
<div class="text-right">
          <form:button class='btn btn-primary'>수정완료</form:button>
          <a
href='${root}board/read?board_info_idx=${board_info_idx}&content_idx=${content_idx}&page=${page}'
                                        class="btn btn-info">취소</a>
                    </div>
          </div>
```

3) modify_success.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:set var='root' value='${pageContext.request.contextPath}/'/>
<script>
        alert('수정되었습니다')
        location.href = 
'${root}board/read?board_info_idx=${modifyContentBean.content_board_idx}&content_idx=${modifyContentBean
.content_idx}&page=${page}'
</script>
```

(5) main.jsp의 게시판 4개 반복해서 표현



1자유게시판

글번호	제목	작성날짜
22	ddd	2020-10-07
21	dd	2020-10-07
20	aaa	2020-10-07
2	bbbb	2020-10-07
1	aaa	2020-10-07

2게시판

글번호	제목	작성날짜
523	테스트용	2020-10-07
522	테스트용	2020-10-07
521	테스트용	2020-10-07
520	테스트용	2020-10-07
519	테스트용	2020-10-07

3게시판

글번호	제목	작성날짜

4게시판

글번호	제목	작성날짜

1) MainService.java 생성

```
@Service
public class MainService {

        @Autowired
        private BoardDao boardDao;

        public List<ContentBean> getMainList(int board_info_idx){
                RowBounds rowBounds = new RowBounds(0, 5);
                return boardDao.getContentList(board_info_idx, rowBounds);
        }
}
```

2) MainController.java

```
@Controller
public class MainController {

        @Autowired
        private MainService mainService;

        @Autowired
        private TopMenuService topMenuService;

        @GetMapping("/main")
        public String main(Model model) {

                ArrayList<List<ContentBean>> list = new ArrayList<List<ContentBean>>();

                for(int i = 1 ; i <= 4 ; i++) {
                        List<ContentBean> list1 = mainService.getMainList(i);
                        list.add(list1);
```

```
                    }
                    model.addAttribute("list", list);

                    List<BoardInfoBean> board_list = topMenuService.getTopMenuList();
                    model.addAttribute("board_list", board_list);

                    return "main";
          }
}
```

3) main.jsp

```
<c:import url="/WEB-INF/views/include/top_menu.jsp"/>

<!-- 게시판 미리보기 부분 -->
<div class="container" style="margin-top:100px">
<div class="row">
          <c:forEach var='sub_list' items="${list }" varStatus="idx">
<div class="col-lg-6" style="margin-top:20px">
          <div class="card shadow">
                    <div class="card-body">
                              <h4 class="card-title">${board_list[idx.index].board_info_name }</h4>
                              <table class="table table-hover" id='board_list'>
                                        <thead>
                                                  <tr>
                                                            <th class="text-center w-25">글번호</th>
                                                            <th>제목</th>
                                                            <th class="text-center w-25 d-none
d-xl-table-cell">작성날짜</th>
                                                  </tr>
                                        </thead>
                                        <tbody>
                                                  <c:forEach var='obj' items='${sub_list }'>
                                                  <tr>
                                                            <td class="text-center">${obj.content_idx }</td>
                                                            <th><a href='${root
}board/read?board_info_idx=${board_list[idx.index].board_info_idx}&content_idx=${obj.content_idx}&page=1'>${o
bj.content_subject }</a></th>
                                                            <td class="text-center d-none
d-xl-table-cell">${obj.content_date }</td>
                                                  </tr>
                                                  </c:forEach>
                                        </tbody>
                              </table>

                              <a href="${root }board/main?board_info_idx=${board_list[idx.index].board_info_idx}"
class="btn btn-primary">더보기</a>
                    </div>
          </div>
</div>
</c:forEach>
          </div>
</div>

<c:import url="/WEB-INF/views/include/bottom_info.jsp"/>

</body>
```