KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

영속,비즈니스 계층의 CRUD 구현

[KB] IT's Your Life

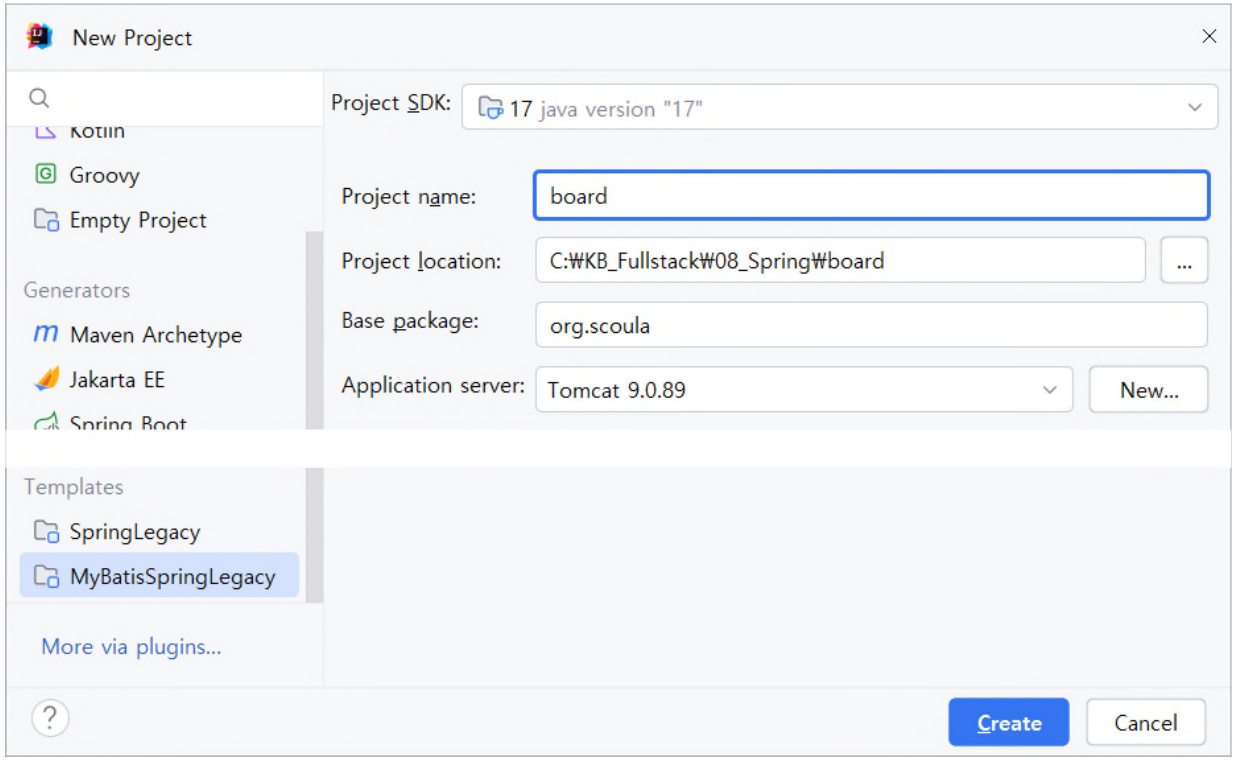
KB 국민은행

프로젝트 만들기

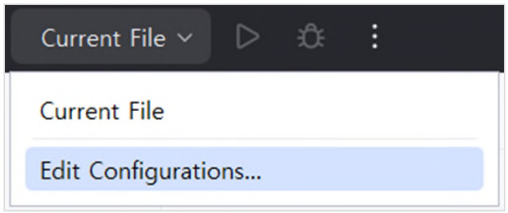
- template: MyBatisSpringLegacy
- project name: board

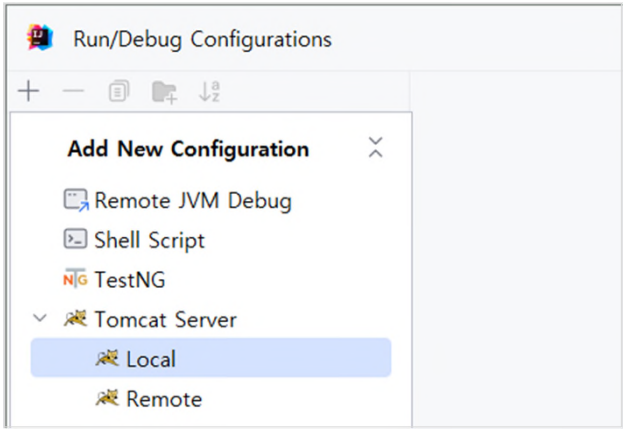
settings.gradle

```
rootProject.name = 'board'
```

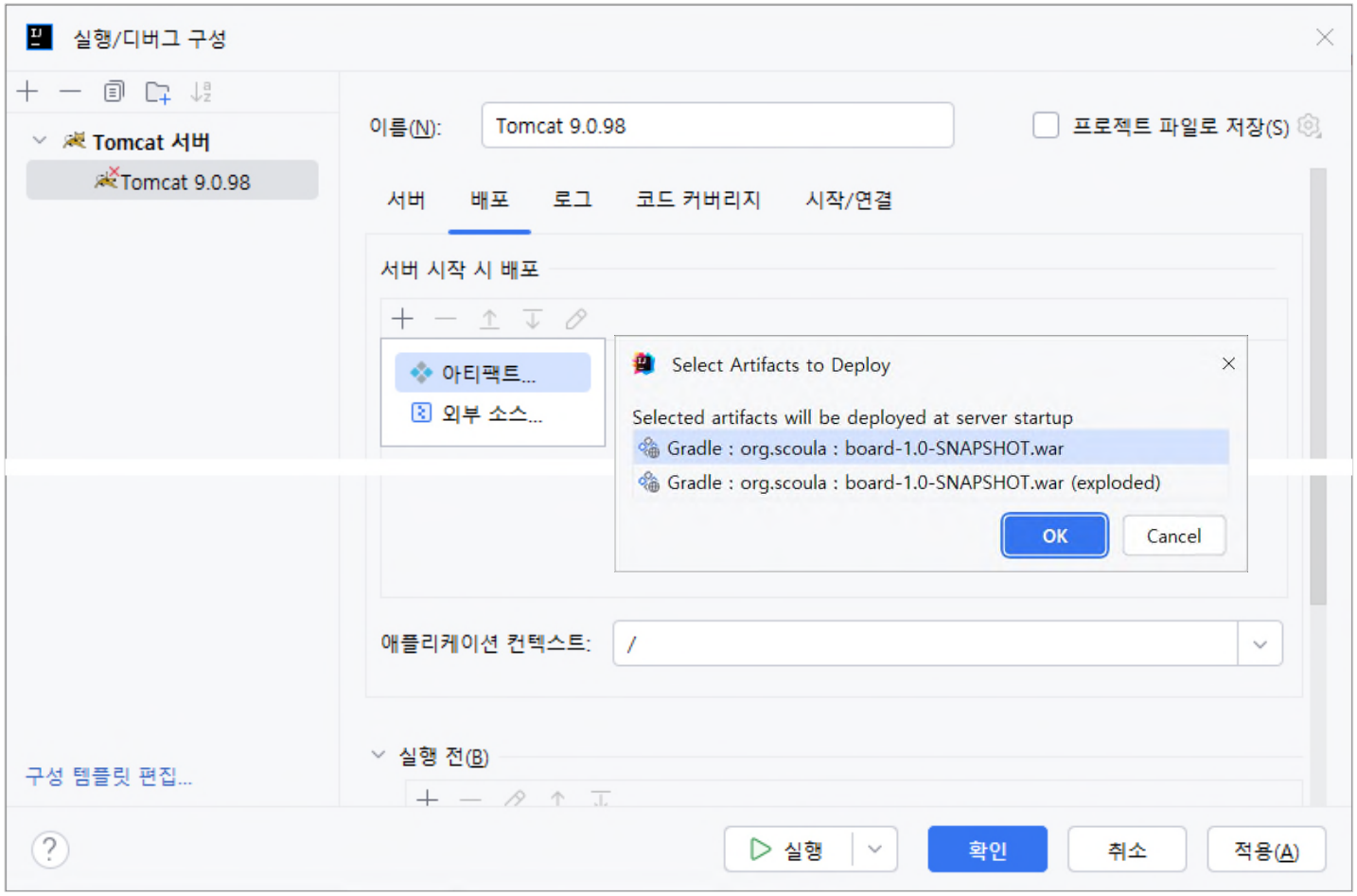


Tomcat 설정


Current File
Current File
Edit Configurations...

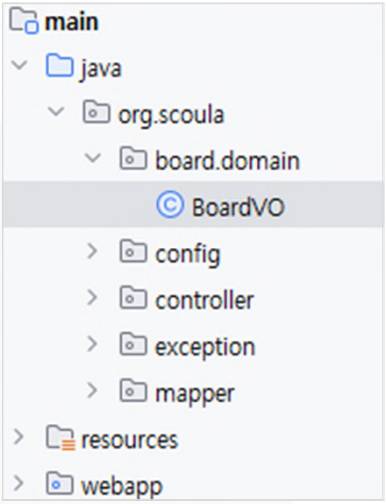

Run/Debug Configurations
Add New Configuration
Remote JVM Debug
Shell Script
TestNG
Tomcat Server
Local
Remote

Tomcat 설정


실행/디버그 구성
Tomcat 서버
Tomcat 9.0.98
이름(N): Tomcat 9.0.98
프로젝트 파일로 저장(S)
서버
배포
로그
코드 커버리지
시작/연결
서버 시작 시 배포
아티팩트...
외부 소스...
Select Artifacts to Deploy
Selected artifacts will be deployed at server startup
Gradle : org.scoula : board-1.0-SNAPSHOT.war
Gradle : org.scoula : board-1.0-SNAPSHOT.war (exploded)
OK
Cancel
애플리케이션 컨텍스트: /
실행 전(B)
구성 템플릿 편집...
실행
확인
취소
적용(A)

database.sql – 테이블 만들기

```
DROP TABLE IF EXISTS  tbl_board;

CREATE TABLE tbl_board (
		no				INTEGER AUTO_INCREMENT PRIMARY KEY,
		title			VARCHAR(200) NOT NULL,
		content			TEXT,
		writer			VARCHAR(50) NOT NULL,
		reg_date		DATETIME DEFAULT CURRENT_TIMESTAMP,
		update_date		DATETIME DEFAULT CURRENT_TIMESTAMP
);

INSERT INTO tbl_board(title, content, writer)
VALUES
		('테스트 제목1', '테스트 내용1', 'user00'),
		('테스트 제목2', '테스트 내용2', 'user00'),
		('테스트 제목3', '테스트 내용3', 'user00'),
		('테스트 제목4', '테스트 내용4', 'user00'),
		('테스트 제목5', '테스트 내용5', 'user00');


SELECT * FROM tbl_board;
```



BoardVO.java

```
package org.scoula.board.domain;

import java.util.Date;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class BoardVO {
   private Long no;
   private String title;
   private String content;
   private String writer;
   private Date regDate;
   private Date updateDate;
}
```

VO 객체의 프로퍼티명과 테이블 컬럼명의 불일치

- updateDate → update_date

- MyBatis 설정 파일에서 설정

resources/mybatis-config.xml

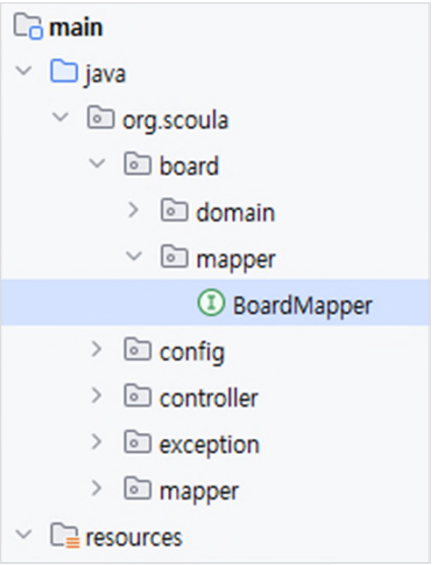
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration
      PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
      "http://mybatis.org/dtd/mybatis-3-config.dtd">

<configuration>
   <settings>
      <setting name="mapUnderscoreToCamelCase" value="true" />
   </settings>

</configuration>
```

MyBatisSpringLegacy 템플릿에 반영

Mapper 인터페이스와 Mapper XML

BoardMapper.java

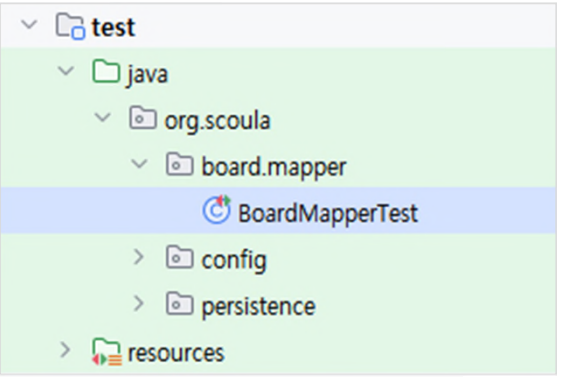
```
package org.scoula.board.mapper;

import java.util.List;

import org.apache.ibatis.annotations.Select;
import org.scoula.board.domain.BoardVO;

public interface BoardMapper {

  @Select("select * from tbl_board order by no desc")
  public List<BoardVO> getList();

}
```



RootConfig.java

```
…

@Configuration
@MapperScan(basePackages  = {"org.scoula.board.mapper"})
public class RootConfig {
                …
}
```

test :: BoardMapper.java




```
package org.scoula.board.mapper;
…

@ExtendWith(SpringExtension.class)
@ContextConfiguration(classes = { RootConfig.class })
@Log4j2
public class BoardMapperTest {

 @Autowired
 private BoardMapper mapper;

 @Test
 @DisplayName("BoardMapper의 목록 불러오기")
 public void getList() {
                for(BoardVO board : mapper.getList()) {
                                log.info(board);
                }
 }
}
```

...

INFO jdbc.sqlonly(sqlOccurred:228) - **select * from tbl_board order by no desc**

INFO org.scoula.board.mapper.BoardMapperTest(getList:28) - BoardVO(no=5, title=테스트 제목5, content=테스트 내용5, writer=user00, regDate=Wed Jan 15 11:53:10 KST 2025, updateDate=Wed Jan 15 11:53:10 KST 2025)

INFO org.scoula.board.mapper.BoardMapperTest(getList:28) - BoardVO(no=4, title=테스트 제목4, content=테스트 내용4, writer=user00, regDate=Wed Jan 15 11:53:10 KST 2025, updateDate=Wed Jan 15 11:53:10 KST 2025)

INFO org.scoula.board.mapper.BoardMapperTest(getList:28) - BoardVO(no=3, title=테스트 제목3, content=테스트 내용3, writer=user00, regDate=Wed Jan 15 11:53:10 KST 2025, updateDate=Wed Jan 15 11:53:10 KST 2025)

INFO org.scoula.board.mapper.BoardMapperTest(getList:28) - BoardVO(no=2, title=테스트 제목2, content=테스트 내용2, writer=user00, regDate=Wed Jan 15 11:53:10 KST 2025, updateDate=Wed Jan 15 11:53:10 KST 2025)

INFO org.scoula.board.mapper.BoardMapperTest(getList:28) - BoardVO(no=1, title=테스트 제목1, content=테스트 내용1, writer=user00, regDate=Wed Jan 15 11:53:10 KST 2025, updateDate=Wed Jan 15 11:53:10 KST 2025)

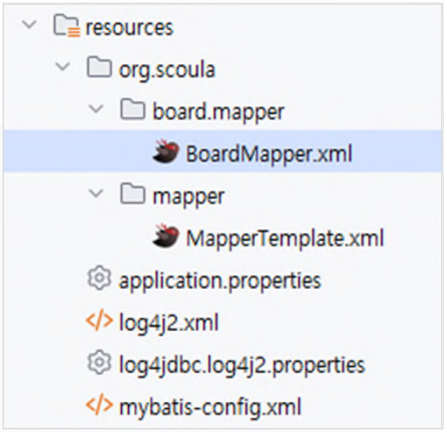
Mapper XML

- 복잡한 쿼리인 경우 @Select()로 구성하는 것은 매우 어려움
- resources 영역에 쿼리를 xml로 작성하여 Mapper 인터페이스와 연계
- Mapper 인터페이스의 패키지 경로 동일하게 동일 파일명으로 작성

BoardMapper.java

```
package org.scoula.board.mapper;

import java.util.List;

import org.apache.ibatis.annotations.Select;
import org.scoula.board.domain.BoardVO;

public interface BoardMapper {

 // @Select("select * from tbl_board")
 public List<BoardVO> getList();

}
```

resources :: BoardMapper.xml (MapperTemplate.xml을 복사해서 생성)




```
<?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="org.scoula.board.mapper.BoardMapper">

        <select id="getList" resultType="org.scoula.board.domain.BoardVO">
                <![CDATA[
                        select * from tbl_board
                        order by no desc
                ]]>
        </select>

</mapper>
```

- <![CDATA[SQL 문자열]]>
  - Compiled Data: 이미 컴파일한 데이터임을 나타냄
    - 문자열 속 <, > 태그 글자가 무시됨
  - 태그 문자가 없다면 생략 가능

Type Alias

<select id="getList" resultType="org.scoula.board.domain.BoardVO">

→ mybatis-config.xml에 단축표현을 위한 설정

<select id="getList" resultType="BoardVO">

resources :: /mybatis-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration
      PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
      "http://mybatis.org/dtd/mybatis-3-config.dtd">

<configuration>
   <settings>
      <setting name="mapUnderscoreToCamelCase" value="true" />
   </settings>

   <typeAliases>
      <package name="org.scoula.board.domain" />
   </typeAliases>

</configuration>
```

resources :: BoardMapper.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="org.scoula.board.mapper.BoardMapper">

                <select id="getList" resultType="BoardVO">
                                <![CDATA[
                                select * from tbl_board
                                ]]>
                </select>

</mapper>
```

BoardMapper.java – get(select) 처리

```
package org.scoula.board.mapper;

import java.util.List;

import org.apache.ibatis.annotations.Select;
import org.scoula.board.domain.BoardVO;

public interface BoardMapper {

  public List<BoardVO> getList();

  public BoardVO get(Long no);
}
```

resources :: BoardMapper.xml – get(select) 처리

```
...
<mapper namespace="org.scoula.board.mapper.BoardMapper">
  ...


        <select id="get" resultType="org.scoula.board.domain.BoardVO">
                select * from tbl_board where no = #{no}
        </select>

</mapper>
```

test :: BoardMapper.java – get(select) 처리

```
...
public class BoardMapperTests {
 ...

 @Test
 @DisplayName("BoardMapper의 게시글 읽기")
 public void get() {
                // 존재하는 게시물 번호로 테스트
                BoardVO board = mapper.get(1L);

                log.info(board);

 }
}
```

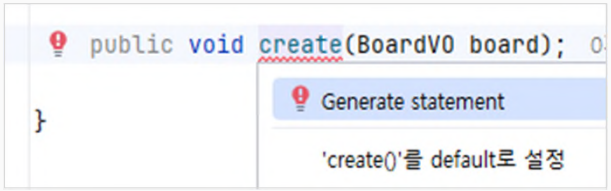
```
...
 INFO  jdbc.sqlonly(sqlOccurred:228) - select * from tbl_board where no = 1

 INFO  org.scoula.board.mapper.BoardMapperTest(get:38) - BoardVO(no=1, title=테스트 제목1, content=테스트 내용1, writer=user00, regDate=Wed Jan 15 11:53:10 KST 2025, updateDate=Wed Jan 15 11:53:10 KST 2025)
...
```

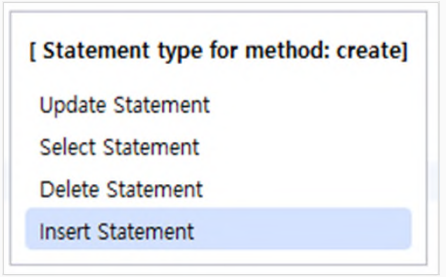
create(insert) 처리

- 시퀀스로 PK가 자동으로 정해지는 경우
- PK 처리 방식
  - insert만 처리되고 생성된 PK 값을 알 필요가 없는 경우
  - insert 문이 실행되고 생성된 PK 값을 알아야 하는 경우
    → insert후 그 키로 다른 테이블에 FK로 후속 작업을 해야 하는 경우
    예) 첨부파일 저장

BoardMapper.java

```
package org.scoula.board.mapper;

import java.util.List;

import org.apache.ibatis.annotations.Select;
import org.scoula.board.domain.BoardVO;

public interface BoardMapper {

  public List<BoardVO> getList();

  public BoardVO get(Long no);

  public void create(BoardVO board);

}
```





resources :: BoardMapper.xml

```
...
<mapper namespace="org.scoula.board.mapper.BoardMapper">
  ...

        <insert id="create">
                insert into tbl_board (title, content, writer)
                values (#{title}, #{content}, #{writer})
        </insert>

</mapper>
```

test :: BoardMapper.java

```
...
public class BoardMapperTests {
  ...
  @Test
  @DisplayName("BoardMapper의 새글 작성")
  public void create() {

    BoardVO board = new BoardVO();
    board.setTitle("새로 작성하는 글");
    board.setContent("새로 작성하는 내용");
    board.setWriter("user0");

    mapper.create(board);

    log.info(board);
  }
}
```

INFO jdbc.sqlonly(sqlOccurred:228) - **insert into tbl_board (title, content, writer) values ('새로 작성하는 글', '새로 작성하는 내용', 'user0')**

INFO org.scoula.board.mapper.BoardMapperTest(create:53) - BoardVO(**no=null**, title=새로 작성하는 글, content=새로 작성하는 내용, writer=user0, regDate=null, updateDate=null)

<selectKey>

- SQL이 실행되기 전에 별도의 PK값 등을 얻기 위해서 사용
- order='before' 를 이용해서 insert구문이 실행되기 전에 호출
- keyProperty를 통해 BoardVO의 no값으로 세팅

속성	설명
keyProperty	PK 값을 읽어 설정할 vo객체의 프로퍼티명
keyColumn	테이블의 PK 컬럼명
resultType	결과의 타입. vo 객체의 프로퍼티 타입 지정
order	selectKey 구문을 언제 실행할 지 지정 - BEFORE: insert 문 실행 전에 selectKey 구문 실행 - AFTER: insert 문 실행 후에 selectKey 구문 실행
statementType	STATEMENT, PREPARED 또는 CALLABLE중 하나를 선택. 디폴트는 PREPARED

resources :: BoardMapper.xml

```
...
<mapper namespace="org.scoula.board.mapper.BoardMapper">
  ...

        <insert id="create">
                insert into tbl_board (title, content, writer)
                values (#{title}, #{content}, #{writer})

                <selectKey resultType="Long" keyProperty="no" keyColumn="no" order="AFTER">
                        SELECT LAST_INSERT_ID()
                </selectKey>
        </insert>
</mapper>
```

test :: BoardMapper.java

```
...
public class BoardMapperTests {
  ...
  @Test
  @DisplayName("BoardMapper의 새글 작성")
  public void create() {

    BoardVO board = new BoardVO();
    board.setTitle("새로 작성하는 글");
    board.setContent("새로 작성하는 내용");
    board.setWriter("user0");

    mapper.create(board);

    log.info(board);
  }
}
```

```
INFO  jdbc.sqlonly(sqlOccurred:228) - insert into tbl_board (title, content, writer) values ('새로 작성하는 글', '새로 작성하는 내용', 'user0')

INFO  jdbc.sqlonly(sqlOccurred:228) - SELECT LAST_INSERT_ID()

INFO  org.scoula.board.mapper.BoardMapperTest(create:53) - BoardVO(no=7, title=새로 작성하는 글, content=새로 작성하는 내용, writer=user
0, regDate=null, updateDate=null)
```

- update 처리

BoardMapper.java - update 처리

```
package org.scoula.board.mapper;

import java.util.List;

import org.apache.ibatis.annotations.Select;
import org.scoula.board.domain.BoardVO;

public interface BoardMapper {

  public List<BoardVO> getList();

  public BoardVO get(Long no);

  public void create(BoardVO board);

  public int update(BoardVO board);
}
```

resources :: BoardMapper.xml – update 처리

```
...
<mapper namespace="org.scoula.board.mapper.BoardMapper">
  ...


        <update id="update">
                update tbl_board set
                        title = #{title},
                        content = #{content},
                        writer = #{writer},
                        update_date = now()
                where no = #{no}
        </update>

</mapper>
```

test :: BoardMapper.java - update 처리

```
...
public class BoardMapperTests {
  ...
  @Test
  @DisplayName("BoardMapper의 글 수정")
  public void update() {

    BoardVO board = new BoardVO();
    board.setNo(5L);
    board.setTitle("수정된 제목");
    board.setContent("수정된 내용");
    board.setWriter("user00");

    int count = mapper.update(board);

    log.info("UPDATE COUNT: " + count);
  }
}
```

```
 INFO  jdbc.sqlonly(sqlOccurred:228) - update tbl_board set title = '수정된 제목', content = '수정된 내용', writer = 'user00', update_date = now()
where no = 5

 INFO  org.scoula.board.mapper.BoardMapperTest(update:68) - UPDATE COUNT: 1
```

- delete 처리

BoardMapper.java – delete 처리

```
package org.scoula.board.mapper;

import java.util.List;

import org.apache.ibatis.annotations.Select;
import org.scoula.board.domain.BoardVO;

public interface BoardMapper {
  public List<BoardVO> getList();

  public BoardVO get(Long no);

  public void create(BoardVO board);

  public int update(BoardVO board);

  public int delete(Long no);
}
```

resources :: BoardMapper.xml – delete 처리

```
...
<mapper namespace="org.scoula.board.mapper.BoardMapper">
  ...

        <delete id="delete">
                delete from tbl_board where no = #{no}
        </delete>

</mapper>
```

test :: BoardMapper.java – delete 처리

```
...
public class BoardMapperTests {
  ...
  @Test
  @DisplayName("BoardMapper의 글 삭제")
  public void delete() {

                log.info("DELETE COUNT: " + mapper.delete(3L));

  }
}
```

```
INFO  jdbc.sqlonly(sqlOccurred:228) - delete from tbl_board where no = 3

INFO  org.scoula.board.mapper.BoardMapperTest(delete:74) - DELETE COUNT: 1
```