

2025년 상반기 K-디지털 트레이닝

# 영속,비즈니스 계층의 CRUD 구현

---

[KB] IT's Your Life

### ✓ 다음과 같이 프로젝트를 만드세요.

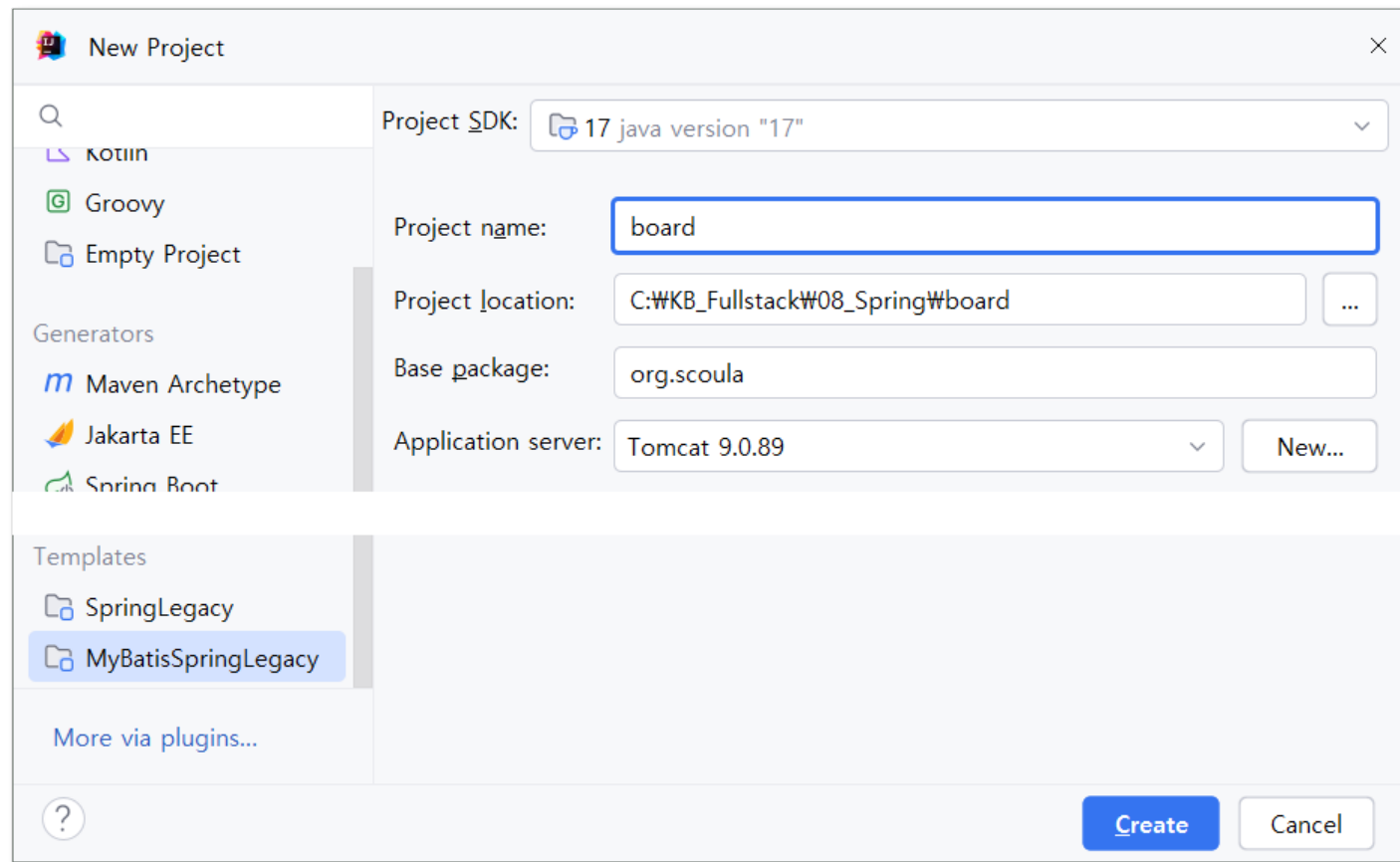
- template: MyBatisSpringLegacy
- project name: board
  
- settings.gradle에서 프로젝트명 변경
- Tomcat 설정

## ✓ 프로젝트 만들기

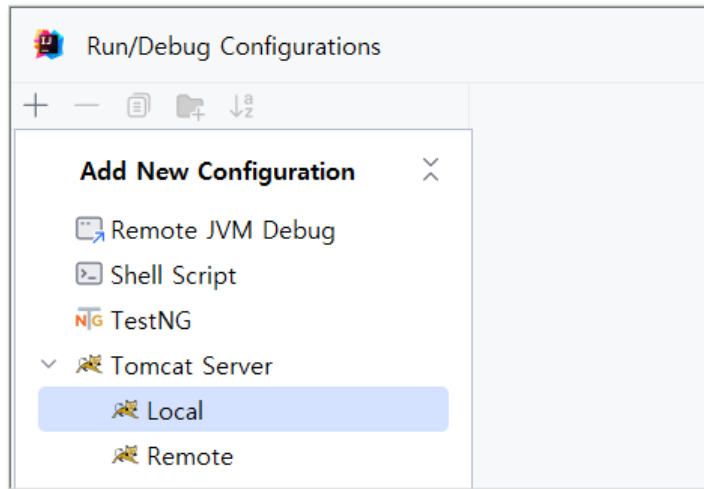
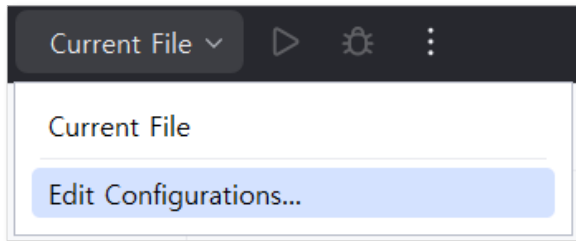
- template: MyBatisSpringLegacy
- project name: board

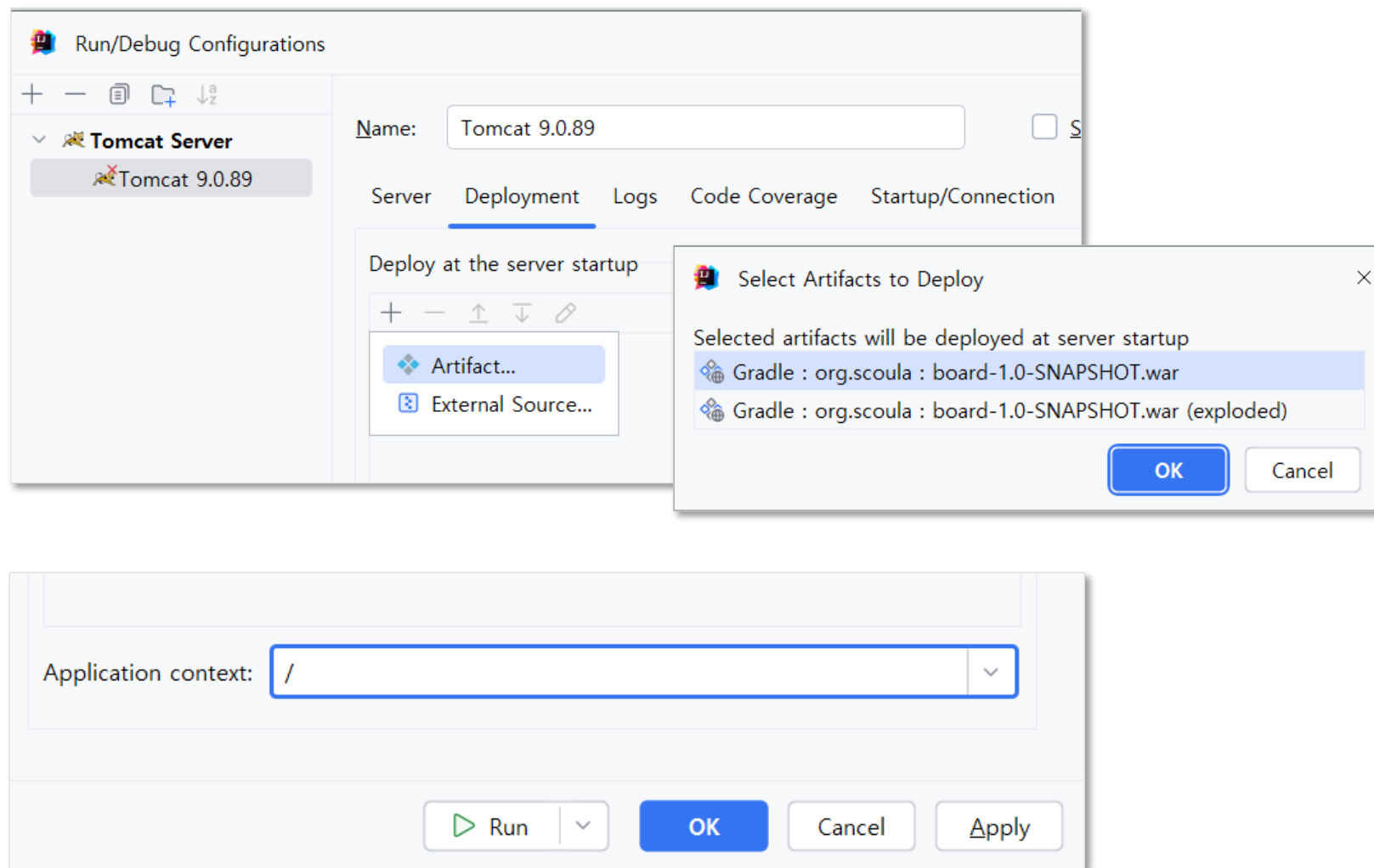
 settings.gradle

rootProject.name = 'board'



## ✓ Tomcat 설정





### ✓ 게시판용 테이블을 생성하고, 5개의 테스트 데이터를 추가하세요.

#### ○ 컬럼

▪	no		PK
▪	title		제목
▪	content	본문	
▪	writer		작성자
▪	reg_date	등록일	
▪	update_date	수정일	

## board.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;
```

- ✓ tbl\_board 테이블을 위한 BoardVO 클래스를 정의하세요.
  - 패키지: org.scoula.board.domain



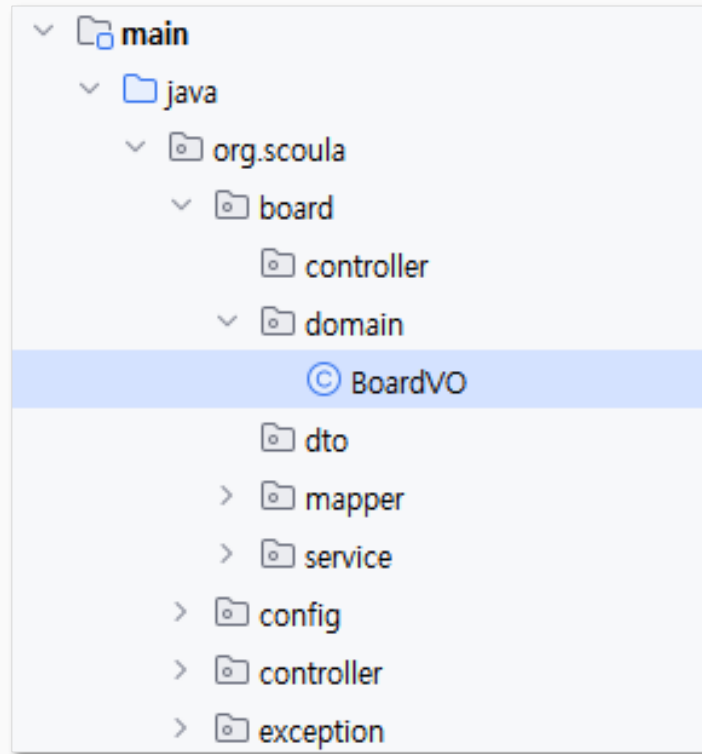
## BoardVO.java

```
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;
}
```



- ✓ **MyBatis 설정 파일에 mapUnderscoreToCamelCase 속성과 BoardVO에 대한 타입 앨리어스 속성을 설정하세요.**

## 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>

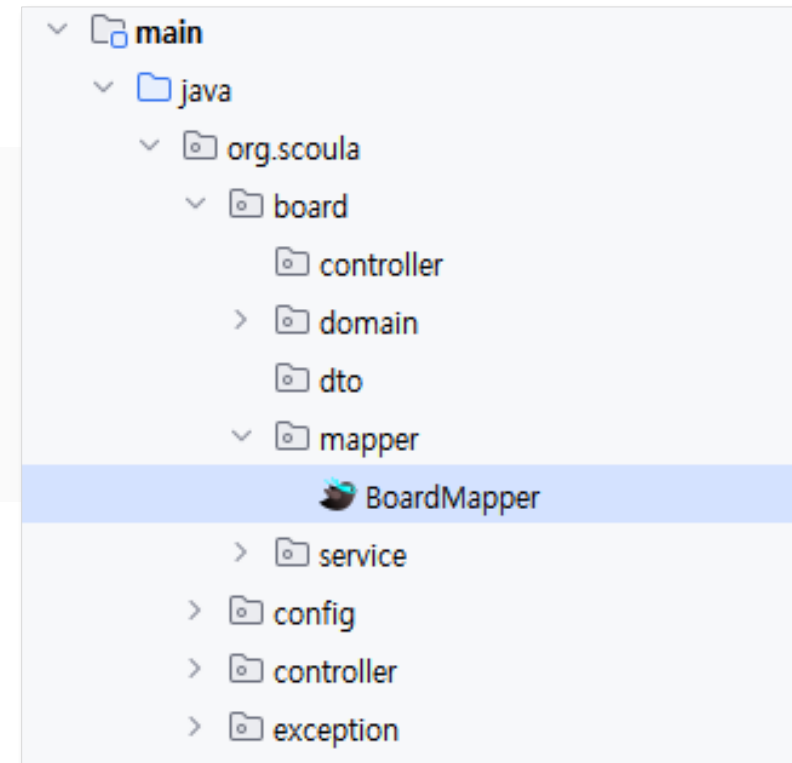
  <typeAliases>
    <package name="org.scoula.board.domain" />
  </typeAliases>
</configuration>
```

### ✓ MyBatis 연동을 위한 기본 구조를 추가하세요.

- BoardMapper 인터페이스
  - 패키지: org.scoula.board.mapper
- RootConfig에 Mapper 스캔 정보 등록
- BoardMapper.xml 파일

## BoardMapper.java

```
package org.scoula.board.mapper;  
  
public interface BoardMapper {  
  
}
```



## RootConfig.java

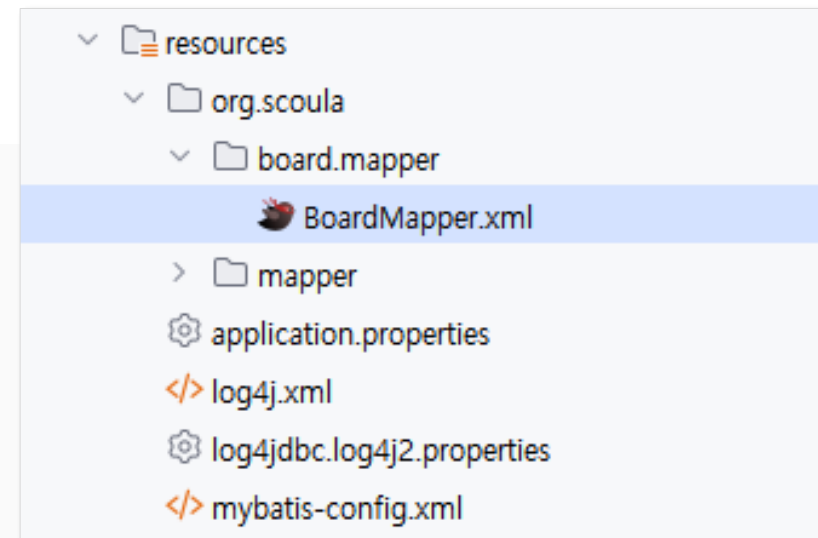
```
...  
  
@Configuration  
@MapperScan(basePackages = {"org.scoula.board.mapper"})  
public class RootConfig {  
    ...  
}
```

## 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">

</mapper>
```



- ✓ 게시판의 목록을 얻는 getList() 메서드를 인터페이스와 xml 매퍼에 추가하고, 단위 테스트를 진행하세요.

## 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();

}
```



## 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
      order by no desc
    ]]>
  </select>

</mapper>
```

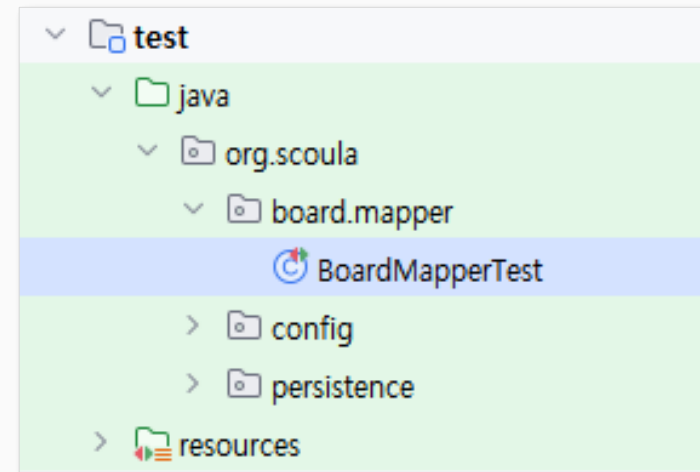
## test :: BoardMapper.java

```
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);
        }
    }
}
```



- ✓ 게시판의 글 하나를 얻는 get(Long no) 메서드를 인터페이스와 xml 매퍼에 추가하고, 단위 테스트를 진행하세요.

## 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="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 : org.scoula.board.mapper.BoardMapperTests - BoardVO(no=1, title=테스트 제목, content=테스트 내용, writer=user00, regdate=Wed May 10 15:33:24 KST 2023, updateDate=Wed May 10 15:33:24 KST 2023)
```

- ✓ 게시판의 글을 추가하는 `int create(BoardVO board)` 메서드를 인터페이스와 xml 매퍼에 추가하고, 단위 테스트를 진행하세요.
  - 글을 추가할 때 생성된 PK를 전달한 객체에 설정함

## 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})

        <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 - insert into tbl\_board (title, content, writer) values ('새로 작성하는 글', '새로 작성하는 내용', 'user0')

...

INFO : org.scoula.board.mapper.BoardMapperTests - BoardVO(no=10, title=새로 작성하는 글, content=새로 작성하는 내용, writer=user0, regdate=null, updateDate=null)

- ✓ 게시판의 글을 수정 하는 `int update(BoardVO board)` 메서드를 인터페이스와 xml 매퍼에 추가 하고, 단위 테스트를 진행하세요.
  - 수정항목: title, content, writer
  - `update_date`는 쿼리 실행한 날짜/시간으로 자동 설정

## 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.sqltiming - update tbl\_board set title = '수정된 제목', content = '수정된 내용', writer = 'user00', updateDate = sysdate where no = 5

...

INFO : org.scoula.board.mapper.BoardMapperTests - UPDATE COUNT: 1

- ✓ 게시판의 글을 삭제하는 `int delete(Long no)` 메서드를 인터페이스와 xml 매퍼에 추가하고, 단위 테스트를 진행하세요.

## 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.sqltiming - delete tbl\_board where no = 3

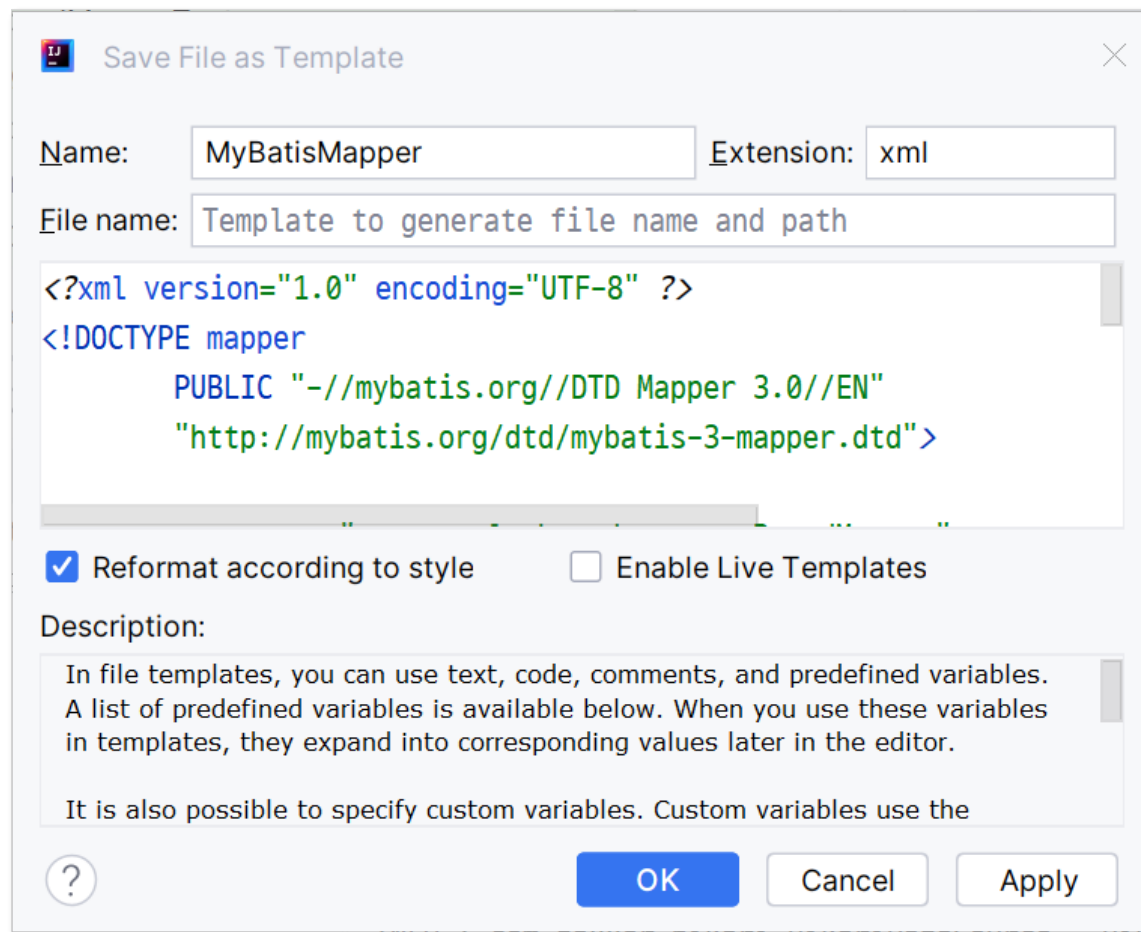
...

INFO : org.scoula.board.mapper.BoardMapperTests - DELETE COUNT: 1

- ✓ 지금까지 작성한 Mapper.xml을 재사용할 수 있도록 수정하여 파일 템플릿으로 저장하세요.

## ✓ Mapper 템플릿

- BoardMapper.xml파일 선택 한 상태에서 File > Save File As Template...



## Mapper 템플릿

```
<?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="${PACKAGE_NAME}.${NAME}">
    <select id="getList" resultType="${VO_NAME}">
        <![CDATA[
            select * from ${TABLE_NAME}
        ]]>
    </select>

    <select id="get" resultType="${VO_NAME}">
        select * from ${TABLE_NAME} where no = #{no}
    </select>
```

## Mapper 템플릿

```
<insert id="create">
  insert into ${TABLE_NAME}()
  values()

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

<update id="update">
  update ${TABLE_NAME} set

  update_date = now()
  where no = #{no}
</update>

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

</mapper>
```