



영속성 계층 (Mapper)

🕒 생성일	@2022년 8월 23일 오전 3:21
🏷 태그	

계층간 전송 데이터 객체 생성 (DTO, VO 객체 생성)

DTO 객체와 VO 객체를 생성한다.

- DTO 객체 : getter/setter가 존재하며, immutable하지 않다.
- VO 객체 : 값 그 자체로, getter만 존재하며, immutable하다.

domain 패키지에 생성한다.

```
@Value
public class BoardVO {
    private Integer bno;
    private String title;
    private String content;
    private String writer;
    private Timestamp insert_ts;
    private Timestamp update_ts;
} // end class
```

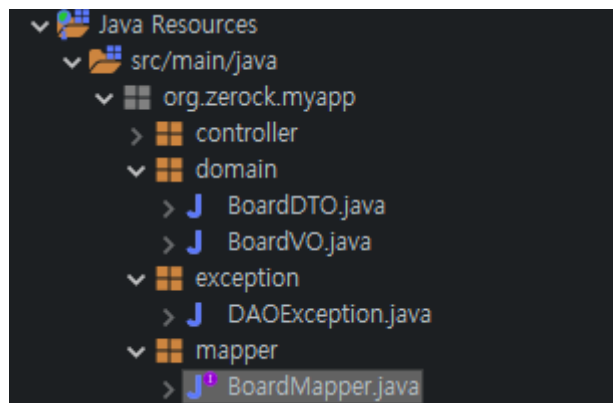
```
@Data
public class BoardDTO {
    private Integer bno;
    private String title;
    private String content;
    private String writer;
    private Timestamp update_ts;
} // end class
```

Mapper 인터페이스와 Mapper XML 생성

mybatis을 DB에 접근하는 방법은 2가지 방법이 존재하는데, 인터페이스에 어노테이션을 사용하는 방법과, 자동실행규칙을 사용한 방법이 있다.

1. 전체 게시물을 조회 - @Select 어노테이션 사용

org.zerock.myapp.mapper 패키이에 BoardMapper Interface 작성



영속성 계층 Mapper를 호출 할 경우, DAOException이란 사용자정의 예외가 checked Exception으로 발생할 수 있도록 throws 한다.

```
public interface BoardMapper {  
  
    // 1. 게시판 테이블의 전체 목록 조회하기  
    @Select(value="SELECT * FROM tbl_board where bno > 0")  
    public abstract List<BoardVO> selectAllList() throws DAOException;  
  
} // end interface
```

사용자 정의 Exception은 Exception클래스를 상속받으며, 생성자로 Exception클래스 생성자가 호출 될 수 있도록 설정한다.

```
public class DAOException extends Exception {

    private static final long serialVersionUID = 1L;

    public DAOException(String message) {
        super(message);
    } // Constructor

    public DAOException(Exception e) {
        super(e);
    } // Constructor

} // end class
```

JUnit5 기준으로 영속성 계층 (모든 게시물 조회)에 대한 TEST CODE 작성

앞서 생성한 BoardMapper에 대한 주입이 필요하여, Setter를 이용한 의존성 주입

```
@Log4j2
@NoArgsConstructor

@ExtendWith(SpringExtension.class)
@ContextConfiguration(locations = {
    "file:src/main/webapp/WEB-INF/spring/root-context.xml"
})

@TestInstance(Lifecycle.PER_CLASS)
@TestMethodOrder(MethodOrderer.OrderAnnotation.class)
public class BoardMapperTests {

    @Setter(onMethod_= {@Autowired})
    private BoardMapper mapper;

    @BeforeAll
    void beforeAll() {
        log.trace("beforeAll() invoked.");

        Objects.requireNonNull(this.mapper);
        log.info("\t+ mapper : {}", this.mapper);
    } // beforeAll

    @Test
    @Order(1)
    @DisplayName("1. testSelectAllList")
    @Timeout(value=3, unit=TimeUnit.SECONDS)
```

```

void testSelectAllList() throws DAOException {
    log.trace("testSelectAllList() invoked.");

    this.mapper.selectAllList().forEach(log::info);
} // testSelectAllList

} // end class

```

```

o.z.m.m.BoardMapperTests.forEach:1541 - BoardVO(bno=158, title=TITLE_158, content=CONTENT_158, writer=writer_158, insert_ts=2022-08-19 20:58:29.0, update_ts=null)
o.z.m.m.BoardMapperTests.forEach:1541 - BoardVO(bno=159, title=TITLE_159, content=CONTENT_159, writer=writer_159, insert_ts=2022-08-19 20:58:29.0, update_ts=null)
o.z.m.m.BoardMapperTests.forEach:1541 - BoardVO(bno=160, title=TITLE_160, content=CONTENT_160, writer=writer_160, insert_ts=2022-08-19 20:58:29.0, update_ts=null)
o.z.m.m.BoardMapperTests.forEach:1541 - BoardVO(bno=161, title=TITLE_161, content=CONTENT_161, writer=writer_161, insert_ts=2022-08-19 20:58:29.0, update_ts=null)
o.z.m.m.BoardMapperTests.forEach:1541 - BoardVO(bno=162, title=TITLE_162, content=CONTENT_162, writer=writer_162, insert_ts=2022-08-19 20:58:29.0, update_ts=null)
o.z.m.m.BoardMapperTests.forEach:1541 - BoardVO(bno=163, title=TITLE_163, content=CONTENT_163, writer=writer_163, insert_ts=2022-08-19 20:58:29.0, update_ts=null)
o.z.m.m.BoardMapperTests.forEach:1541 - BoardVO(bno=164, title=TITLE_164, content=CONTENT_164, writer=writer_164, insert_ts=2022-08-19 20:58:29.0, update_ts=null)
o.z.m.m.BoardMapperTests.forEach:1541 - BoardVO(bno=165, title=TITLE_165, content=CONTENT_165, writer=writer_165, insert_ts=2022-08-19 20:58:29.0, update_ts=null)
o.z.m.m.BoardMapperTests.forEach:1541 - BoardVO(bno=166, title=TITLE_166, content=CONTENT_166, writer=writer_166, insert_ts=2022-08-19 20:58:29.0, update_ts=null)
o.z.m.m.BoardMapperTests.forEach:1541 - BoardVO(bno=167, title=TITLE_167, content=CONTENT_167, writer=writer_167, insert_ts=2022-08-19 20:58:29.0, update_ts=null)
o.z.m.m.BoardMapperTests.forEach:1541 - BoardVO(bno=168, title=TITLE_168, content=CONTENT_168, writer=writer_168, insert_ts=2022-08-19 20:58:29.0, update_ts=null)

```

2. 새로운 게시물 등록 - Mapper XML로 처리

BoardMapper에 추상메소드 선언

-. 어노테이션 Insert를 사용하지 않고, Mapper XML을 생성하여, 쿼리를 추가한다.

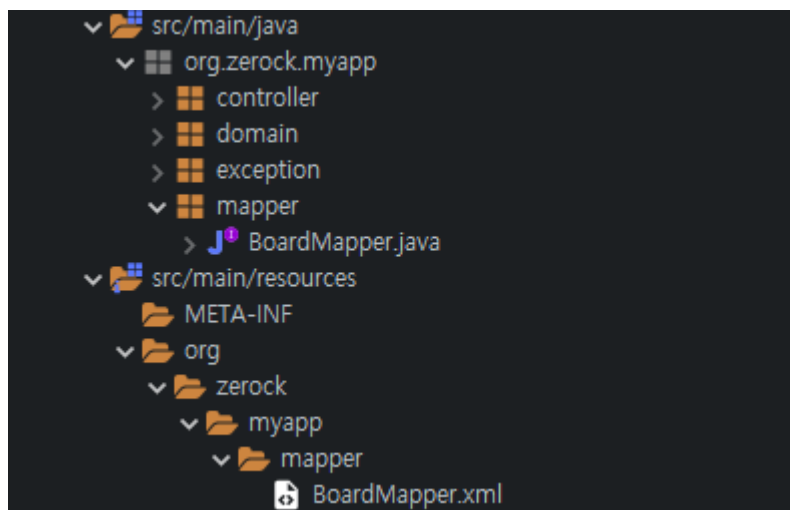
```

// 2. 새로운 게시글을 등록하기 - Mapper XML 파일로 처리
public abstract Integer insert(BoardDTO dto) throws DAOException;

```

Mapper XML 파일 생성 (자동생성규칙 준수)

-. Resources 폴더 내, 동일한 Package 명으로 저장 (파일명도 Interface와 동일)



BoardMapper.xml의 Default 설정 (아래 코드 참고)

-. mapper 태그 내, **namespace** 속성은 Mapper 인터페이스의 FQCN 입력한다.

```
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="org.zerock.myapp.mapper.BoardMapper">

</mapper>
```

BoardMapper.xml에 새로운 게시물 등록에 대한 쿼리를 mybatis를 이용하여 작성

-. insert 태그로 작성

-. id속성은 Mapper 인터페이스의 메소드명과 동일하게 작성한다.

```
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="org.zerock.myapp.mapper.BoardMapper">
  <insert id="insert">
    INSERT INTO tbl_board(title, content, writer)
    VALUES (#{title}, #{content}, #{writer})
  </insert>
</mapper>
```

TEST코드 작성

-. 파라미터로 DTO 객체 전달

```
@Test
@Order(2)
@DisplayName("2. testInsert")
@Timeout(value=3, unit=TimeUnit.SECONDS)
void testInsert() throws DAOException {
  log.trace("testInsert() invoked.");

  BoardDTO dto = new BoardDTO();
  dto.setTitle("TITLE_NEW");
  dto.setContent("CONTENT_NEW");
  dto.setWriter("WRITER_NEW");
}
```

```
log.info("\t+ result : {}", this.mapper.insert(dto));  
} // testInsert
```

3. 기존 게시글 상세 조회 - Mapper XML 파일로 처리

BoardMapper 인터페이스에 추상메소드 선언

-. BoardDTO 객체를 매개변수로 받아, BoardVO객체를 리턴

```
// 3. 기존 게시글 상세 조회하기 - Mapper XML파일로 처리  
public abstract BoardVO select(BoardDTO dto) throws DAOException;
```

BoardMapper.xml에 기존 게시물 상세 조회 작성

-. id속성 값은 BoardMapper 인터페이스 메소드명과 동일

-. resultType은 BoardVO이므로 FQCN을 작성 (Return타입이 BoardVO)

```
<select id="select" resultType="org.zerock.myapp.domain.BoardVO">  
    SELECT * FROM tbl_board WHERE bno = #{bno}  
</select>
```

TEST코드 작성

```
@Test  
@Order(3)  
@DisplayName("3. testSelect")  
@Timeout(value=3, unit=TimeUnit.SECONDS)  
void testSelect() throws DAOException {  
    log.trace("testSelect() invoked.");  
  
    BoardDTO dto = new BoardDTO();  
  
    dto.setBno(300);  
  
    log.info("\t+ BoardVO : {}", this.mapper.select(dto));  
} // testSelect
```

bno	title	content	writer	insert_ts	update_ts
300	TITLE_300	CONTENT_300	writer_300	2022-08-19 20:58:29.0	null

4. 기존 게시물 수정하기 - Mapper XML 파일로 처리

BoardMapper 인터페이스에 추상메소드 선언

```
// 4. 기존 게시물 수정하기 - Mapper XML파일로 처리
public abstract Integer update(BoardDTO dto) throws DAOException;
```

BoardMapper.xml에 기존 게시물 수정하기 작성

```
<update id="update">
    UPDATE tbl_board
    SET
        title=#{title},
        content=#{content},
        writer=#{writer},
        update_ts=current_date
    WHERE
        bno=#{bno}
</update>
```

TEST코드 작성

```
@Test
@Order(4)
@DisplayName("4. testUpdate")
@Timeout(value=3, unit=TimeUnit.SECONDS)
void testUpdate() throws DAOException {
    log.trace("testUpdate() invoked.");

    BoardDTO dto = new BoardDTO();

    dto.setBno(80);
    dto.setTitle("TITLE_UPDATE");
```

```

        dto.setContent("CONTENT_UPDATE");
        dto.setWriter("WRITER_UPDATE");

        log.info("\t+ result : {}", this.mapper.update(dto) == 1);
    } // testUpdate

```

5. 기존 게시물 삭제하기 - @Delete 어노테이션

BoardMapper 인터페이스에 추상메소드 선언

- 전달되는 매개변수가 mybatis의 바인딩변수에 입력될 수 있도록 @Param 어노테이션을 매개변수에 붙인다.

```

// 5. 기존 게시물 삭제하기
@Delete(value="DELETE FROM tbl_board WHERE bno = #{bno}")
public abstract Integer delete(@Param("bno") Integer bno) throws DAOException;

```

TEST코드 작성

```

@Test
@Order(5)
@DisplayName("5. BoardMapper.testDelete")
@Timeout(value=3, unit=TimeUnit.SECONDS)
void testDelete() throws DAOException {
    log.trace("testDelete() invoked.");

    log.info("\t+ result : {}", this.mapper.delete(36));
} // testDelete

```