

2025년 상반기 K-디지털 트레이닝

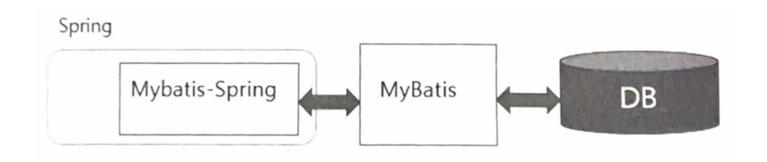
스프링과 MySQL Database 연동

[KB] IT's Your Life



MyBatis

- https://mybatis.org/mybatis-3/
- SQL 매핑 프레임워크



MyBatis 관련 라이브러리 추가

- o spring-jdbc/spring-tx
 - 스프링에서 데이터베이스 처리와 트랜잭션 처리
- mybatis/mybatis-spring
 - MyBatis와 스프링 연동용 라이브러리

build.gradle

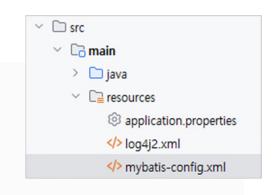
→ gradle sync

SQLSessionFactory

- o SQLSession
 - Connection 생성, SQL 전달, 결과 리턴 등 처리
- SQLSessionFactory
 - SQLSession 객체 생성

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>
</configuration>
```



config.RootConfig.java

```
@Configuration
public class RootConfig {
           @Autowired
           ApplicationContext applicationContext;
@Bean
public SqlSessionFactory sqlSessionFactory() throws Exception {
  SqlSessionFactoryBean sqlSessionFactory = new SqlSessionFactoryBean();
  sqlSessionFactory.setConfigLocation(applicationContext.getResource("classpath:/mybatis-config.xml"));
  sqlSessionFactory.setDataSource(dataSource());
  return (SqlSessionFactory) sqlSessionFactory.getObject();
           @Bean
           public DataSourceTransactionManager transactionManager(){
                       DataSourceTransactionManager manager = new DataSourceTransactionManager(dataSource());
                       return manager;
```

test:: RootConfigTest.java

```
class RootConfigTest {
  @Autowired
  private SqlSessionFactory sqlSessionFactory;
  @Test
  public void testSqlSessionFactory() {
    try (
     SqlSession session = sqlSessionFactory.openSession();
     Connection con = session.getConnection();
      log.info(session);
      log.info(con);
    } catch (Exception e) {
      fail(e.getMessage());
                                        G G B - V O F E O @ 9 @ :
                                   ✓ ✓ RootConfigTest (org.scoula.ci 122ms
                                                                ✓ 테스트 통과: 1 /1개 테스트 - 122ms

✓ testSqlSessionFactory() 122ms

                                                                 INFO org.scoula.config.RootConfigTest(testSqlSessionFactory:46) - org.apache.ibatis/
                                                                  .session.defaults.DefaultSqlSession@52cb4f50
```

INFO org.scoula.config.RootConfigTest(testSqlSessionFactory:47) -

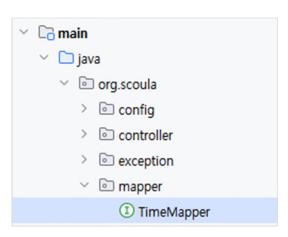
HikariProxyConnection@631621595 wrapping com.mysql.cj.jdbc.ConnectionImpl@4d27d9d

Mapper

- SQL과 그에 대한 처리를 지정하는 역할
- XML과 인터페이스+어노테이션 형태로 작성

Mapper 인터페이스

- o org.scoula.mapper 패키지
 - TimeMapper 인터페이스 추가

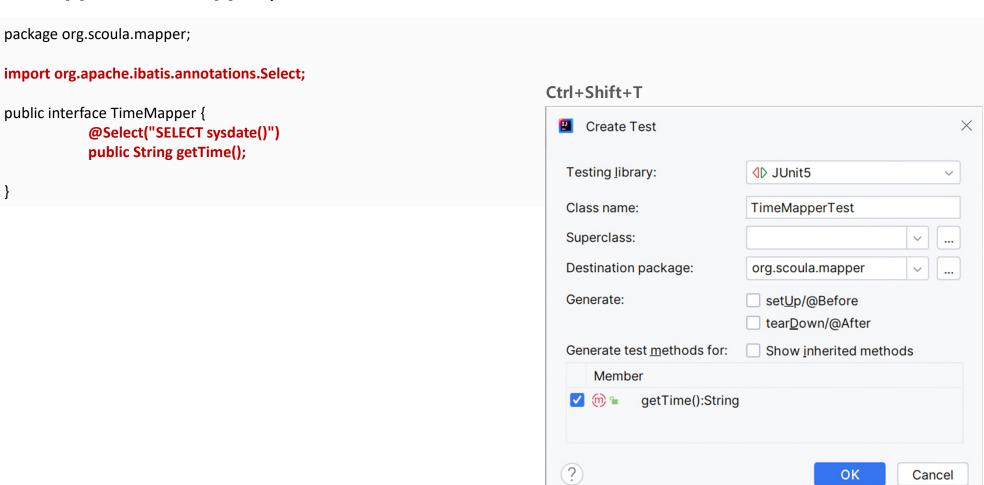


☑ RootConfig.java - Mapper 설정

O @MapperScan

- Mapper 인터페이스를 검색할 패키지 목록 지정
- 해당 인터페이스를 빈으로 등록
- 해당 인터페이스의 구현체를 동적으로 자동 생성

mapper.TimeMapper.java

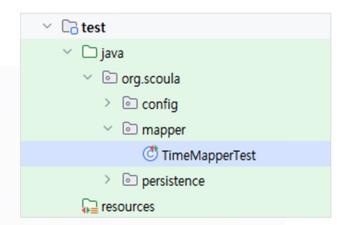



```
@ExtendWith(SpringExtension.class)
@ContextConfiguration(classes = { RootConfig.class })
@Log4j2
public class TimeMapperTest {

@Autowired
    private TimeMapper timeMapper;

@Test
@DisplayName("TimeMapper의 getTime()")
    public void getTime () {
        log.info(timeMapper.getClass().getName());
        log.info(timeMapper.getTime());
    }
}

All TimeMapperTest.getTime ×
```

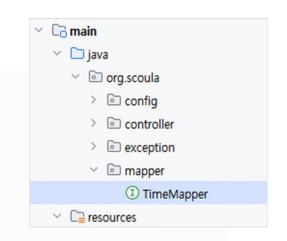


TimeMapper.java

```
package org.scoula.mapper;
import org.apache.ibatis.annotations.Select;

public interface TimeMapper {
         @Select("SELECT sysdate FROM dual")
         public String getTime();

         public String getTime2();
}
```



O getTime2()에 연관시킬 sql은 xml 매퍼로 정의해서 처리

XML 매퍼와 같이 쓰기

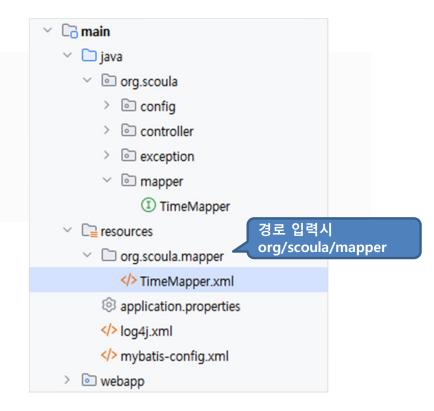
- src/main/resources
 - org/scoula/mapper

--> Mapper 인터페이스와 같은 패키지 경로여야 함

- TimeMapper.xml --> 인터페이스명과 파일명이 같아야 함

```
<?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.mapper.TimeMapper">
</mapper>

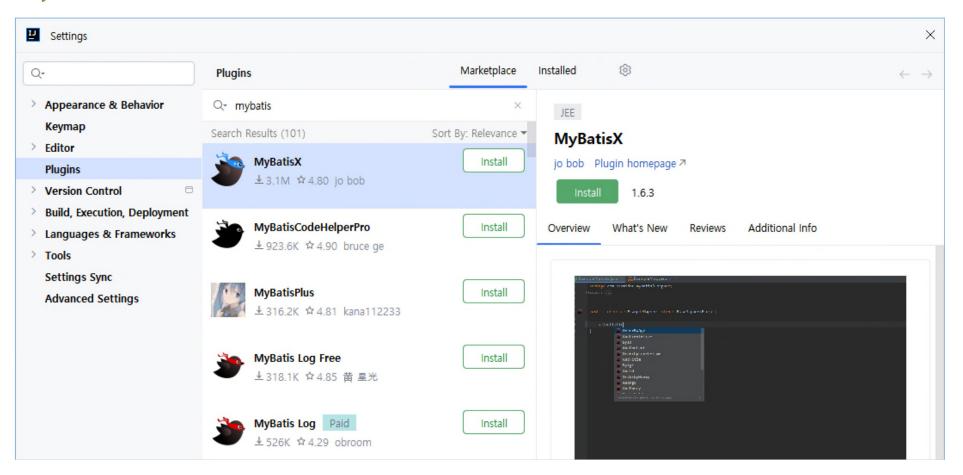
</mapper>
```



✓ resources:: TimeMapper.xml

MyBatisX 플러그인 설치

o mybatis 검색



✓ resources:: TimeMapper.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.mapper.TimeMapper">
</mapper>
```



resources:: TimeMapper.xml



TimeMapper.java

test::TimeMapperTest.java

```
package org.scoula.persistence;
...
public class TimeMapperTest {
...
@Test
@DisplayName("TimeMapper의 getTime2()")
public void getTime2() {

log.info("getTime2");
log.info(timeMapper.getTime2());
}
}
```



JDBC 처리 관련 로그 세부 사항 출력

- PreparedStatement로 SQL 처리시 ?에 실제 치환 값을 알아야 하는 상황이 발생 가능
- log4jdbc-log4j2 라이브러리
 - PreparedStatement로 SQL 처리시 ?에 실제 치환 값을 출력해줌

build.gradle

```
// logging
implementation 'org.slf4j:slf4j-api:2.0.9'
runtimeOnly 'org.slf4j:jcl-over-slf4j:2.0.9'
runtimeOnly 'org.slf4j:slf4j-log4j12:2.0.9'
implementation 'log4j:log4j:1.2.17'
implementation 'org.bgee.log4jdbc-log4j2:log4jdbc-log4j2-jdbc4:1.16'
implementation 'org.apache.logging.log4j:log4j-api:2.0.1'
implementation 'org.apache.logging.log4j:log4j-core:2.0.1'
```

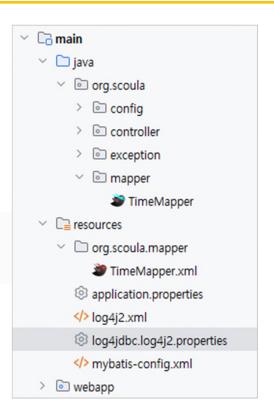
→ gradle sync

log4jdbc-log4j2 설정

- o src/main/resources
 - log4jdbc.log4j2.properties

resources/log4jdbc.log4j2.properties

log4jdbc.spylogdelegator.name=net.sf.log4jdbc.log.slf4j.Slf4jSpyLogDelegator log4jdbc.auto.load.popular.drivers=false log4jdbc.drivers=com.mysql.cj.jdbc.Driver



resources/application.properties

#driver=com.mysql.cj.jdbc.Driver #url=jdbc:mysql://127.0.0.1:3306/scoula_db jdbc.driver=net.sf.log4jdbc.sql.jdbcapi.DriverSpy jdbc.url=jdbc:log4jdbc:mysql://localhost:3306/scoula_db jdbc.username=scoula jdbc.password=1234

☑ 로그의 레벨 설정

- 너무 많은 로그 메시지 출력 → 로그 레벨 이용하여 수정 가능
- 로그레벨
 - FATAL: 가장 크리티컬한 에러가 일어 났을 때만 로깅
 - ERROR: 일반 에러가 일어 났을 때만 로깅
 - WARN: 에러는 아니지만 주의할 필요가 있을 때만 로깅
 - INFO: 일반 정보를 나타낼 때 로깅 (INFO + ERROR)
 - DEBUG: 일반 정보를 상세히 나타낼 때 로깅 (디버깅 정보)
 - TRACE: 경로추적을 위해 사용

test :: resources/log4j2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration>
  <!-- Appender, Layout 설정 -->
  <Appenders>
    <Console name="console" target="SYSTEM OUT">
      <PatternLayout pattern=" %-5level %c(%M:%L) - %m%n"/>
    </Console>
  </Appenders>
  <!-- Logger 설정 -->
  <Loggers>
    <Root level="INFO">
      <AppenderRef ref="console"/>
    </Root>
    <Logger name="org.scoula" level="INFO" additivity="false" >
      <AppenderRef ref="console"/>
    </Logger>
    <Logger name="org.springframework" level="INFO" additivity="false">
      <AppenderRef ref="console"/>
    </Logger>
  </Loggers>
</Configuration>
```

♡ 테스트 로그 출력

```
INFO jdbc.audit(methodReturned:175) - 1. Connection.prepareStatement(SELECT sysdate()) returned net.sf.log4jdbc.sql.jdbcapi.PreparedS tatementSpy@31a3f4de
INFO jdbc.sqlonly(sqlOccurred:228) - SELECT sysdate()
INFO jdbc.sqltiming(sqlTimingOccurred:373) - SELECT sysdate()
{executed in 24 msec}
INFO jdbc.audit(methodReturned:175) - 1. PreparedStatement.execute() returned true
INFO jdbc.resultset(methodReturned:175) - 1. ResultSet.new ResultSet returned
INFO jdbc.audit(methodReturned:175) - 1. PreparedStatement.getResultSet() returned net.sf.log4jdbc.sql.jdbcapi.ResultSetSpy@302edb74
INFO jdbc.resultset(methodReturned:175) - 1. ResultSet.getMetaData() returned com.mysql.cj.jdbc.result.ResultSetMetaData@67c5ac52 - Field level information:
```

com.mysql.cj.result.Field@36417a54[dbName=null,tableName=null,originalTableName=null,columnName=sysdate(),originalColumnName=null,mysqlType=12(FIELD TYPE DATETIME),sqlType=93,flags= BINARY, charsetIndex=63, charsetName=ISO-8859-1]

```
...
|-----|
|sysdate() |
|------|
|2025-01-15 11:19:37 |
|------|
```

☑ 로그의 레벨 설정

- log4jdbc 출력 로그 조절
 - <Logger> 태그 지정

○ 핵심 로그만 출력하기

- jdbc.sqlonly
- → 로그 레벨 info로 설정

테스트용 log4j2.xml

- jdbc.audit
- jdbc.connection
- jdbc.resultset 등
- → 해당 패키지의 로그 레벨을 warn으로 설정

resources/log4j2.xml

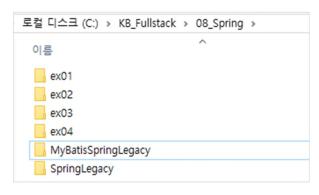
```
<Logger name="jdbc" level="WARN" additivity="false">
      <AppenderRef ref="console"/>
    </Logger>
    <Logger name="jdbc.sqlonly" level="INFO" additivity="false">
      <AppenderRef ref="console"/>
    </Logger>
  </Loggers>
</Configuration>
```

ㅇ 다시 테스트 실행



🛾 프로젝트 템플릿 만들기

○ ex04 → MyBatisSpringLegacy로 복사

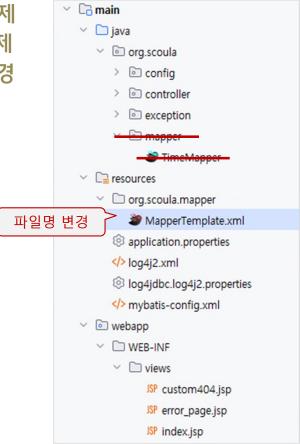


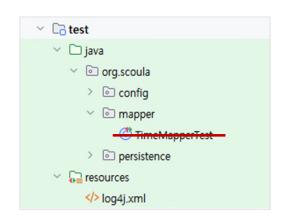
settings.gradle

rootProject.name = "MyBatisSpringLegacy"

🕝 파일 정리

- o mapper 관련 패키지/파일 삭제
- o mapper 관련 테스트 파일 삭제
- TimeMapper.xml 파일명 변경
 - → MapperTemplate.xml





RootConfig.java

resources/org/scoula/mapper/MapperTemplate.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="">
</mapper>
```

💟 프로젝트 템플릿 만들기

File > New Project Setup > Save Project As Template...

