

# MyBatis 스프링 연동하기

🕒 생성일	@2022년 8월 16일 오후 10:52
☰ 태그	

## 1. pom.xml 설정

MyBatis도 프레임워크로, pom.xml에 라이브러리를 추가해주어야 한다.

-. mybatis와 스프링 연동용 라이브러리 추가

```
<!-- https://mvnrepository.com/artifact/org.mybatis/mybatis -->
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis</artifactId>
  <version>3.5.10</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.mybatis/mybatis-spring -->
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis-spring</artifactId>
  <version>2.0.7</version>
</dependency>
```

-. 스프링 데이터베이스 처리와 트랜잭션 처리 라이브러리 추가

-. 스프링 프레임워크관련 라이브러리는 Version이 모두 통일되어야 한다. (동일버전사용)

```
<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-jdbc</artifactId>
  <version>${org.springframework-version}</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-tx -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-tx</artifactId>
  <version>${org.springframework-version}</version>
</dependency>
```

## 2. root-context.xml 작성

SQLSession을 생성하는 SQLSessionFactory 빈 객체로 등록필요

- 속성값으로 DataSource 객체를 주입받는다. HikariDataSource나, mybatisDataSource를 참조한다.
- mybatis-config.xml 속성값으로 지정 : value 속성값으로 classpath: 를 이용한다. resources root를 의미한다.

```
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean" >
  <property name="dataSource" ref="pooledDataSource"></property>
  <!-- <property name="dataSource" ref="hikariDataSource"></property> -->

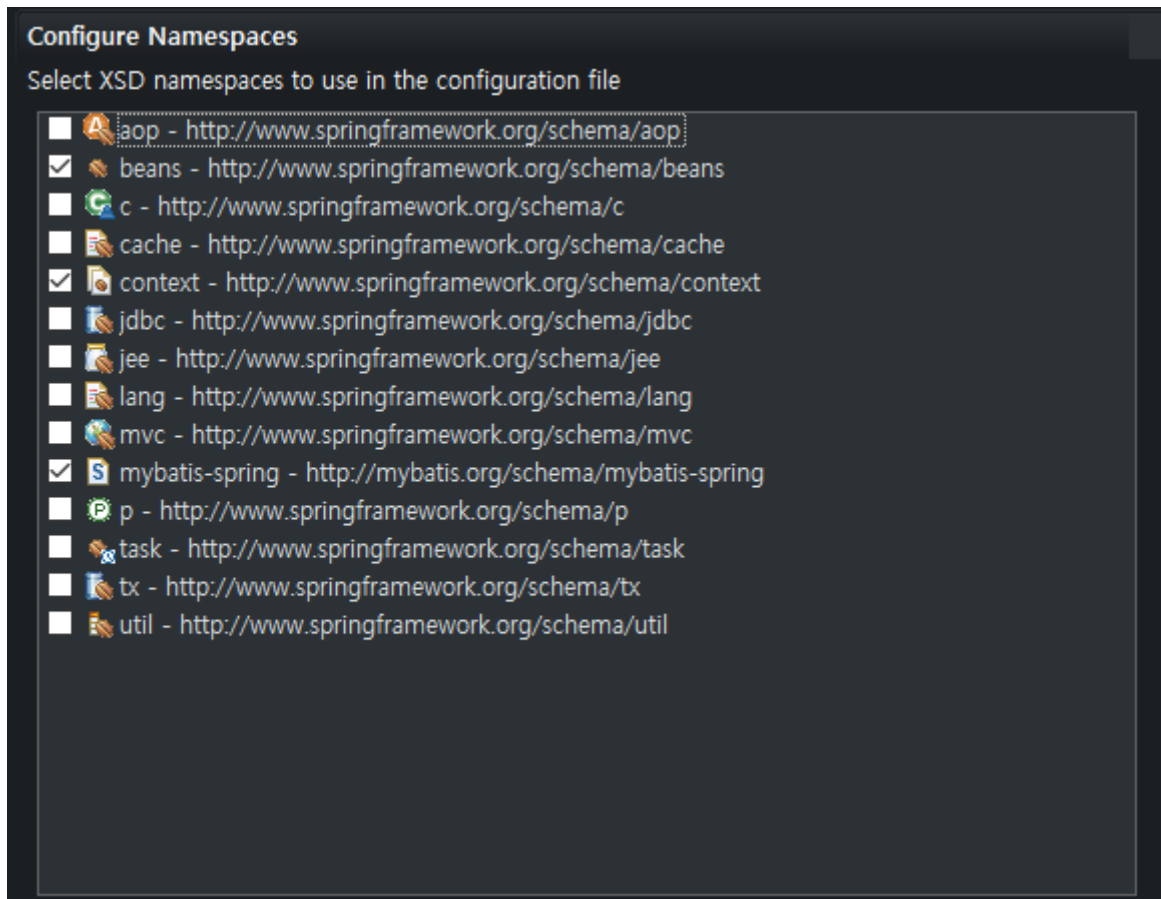
  <property name="configLocation" value="classpath:mybatis-config.xml" ></property>
</bean>

<!-- 만약, 마이바티스 설정파일 내에서, SQL Mapper 파일의 경로를 직접 지정하는 경우에는 설정이 필요 없음 -->
<!-- <property name="mapperLocations" value="classpath:mappers/**/*.xml" >
</property> -->
</bean>
```

## 3. SQL 처리하기 (어노테이션 @Select 이용)

### 1. root-context.xml Namespaces mybatis-spring 체크

- Mapper 인터페이스 경로를 인식하게 <mybatis-spring:scan> 태그를 이용하여 스캔



- mapper 인터페이스는 동일한 패키지에 저장하는 것이 편리하다.

```
<mybatis-spring:scan base-package="org.zerock.myapp.mapper" />
```

## 2. Mapper 인터페이스 작성

- @Select 어노테이션을 사용하여, SQL 쿼리문 value값으로 작성한다.

```
public interface TimeMapper {  
  
    @Select(value="SELECT to_char(current_date, 'yyyy/MM/dd HH24:mi:ss') FROM dual")  
    public abstract String getTime();  
  
    public abstract String getTime2();  
}
```

```
} // end interface
```

### 3. Test 코드 작성

- SqlSessionFactory 객체는 root-context.xml에 빈으로 등록되었기 때문에, Setter 메소드를 통한 의존성 주입을 한다. (의존성 주입이 잘 되었는 지 테스트코드는 생략...)

- sqlSession의 getMapper( ) 메소드를 이용하여, Mapper 클래스 생성

<T> T getMapper(Class<T> type)의 형태로, 매개변수의 클래스 타입을 리턴한다.

```
@Log4j2
@NoArgsConstructor

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

@TestInstance(Lifecycle.PER_CLASS)
@TestMethodOrder(OrderAnnotation.class)
public class Ex02_TimeMapperTests {

    @Setter(onMethod_={@Autowired})
    private SqlSessionFactory sqlSessionFactory;

    @Test
    @Order(1)
    @DisplayName("1. testTime")
    @Timeout(value=3, unit=TimeUnit.SECONDS)
    void testGetTime1() {
        log.trace("testGetTime1() invoked.");

        SqlSession sqlSession = this.sqlSessionFactory.openSession();

        try (sqlSession) {
            TimeMapper mapper = sqlSession.getMapper(TimeMapper.class);
            Objects.requireNonNull(mapper);
            log.info("\t+ mapper : {}", mapper);

            String time = mapper.getTime();
            log.info("\t+ time : {}", time);
        }
    }
}
```

```

    } // try-with-resources

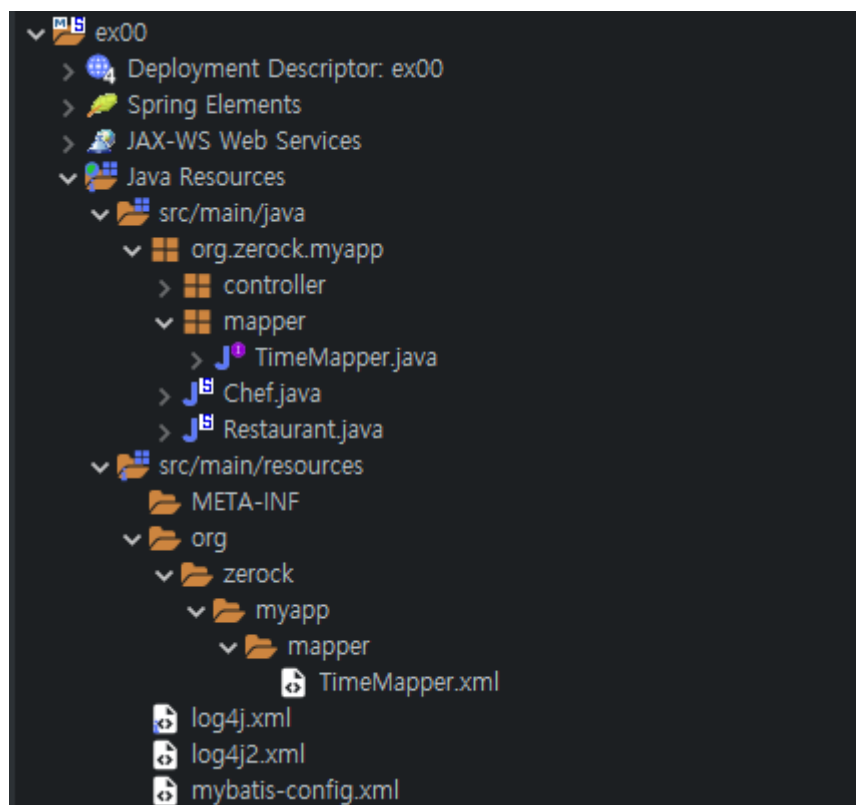
    } // testGetTime1

} // end class

```

## 4. SQL 처리하기 (xml을 이용한 처리 : 자동실행규칙)

자동실행규칙 전제조건



- (1) Mapper Interface의 패키지와 동일한 폴더구조를 Resource폴더 아래 생성
- (2) 위 (1)에서 생성한 폴더구조 아래에, Mapper Interface와 동일한 이름의 Mapper XML 파일생성

(3) 위 (2)에서 생성한 Mapper XML파일의 namespace에는 Mapper Interface의 FQCN을 적는다.

- . 사전에 문서타입 작성 필요

(4) id 속성은 Mapper Interface의 메소드명으로 적는다.

(5) resultType은 반환을 받을 타입의 FQCN을 적는다.

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

<mapper namespace="org.zerock.myapp.mapper.TimeMapper">
  <select
    id="getTime2"
    resultType="java.lang.String">
    SELECT to_char(current_date, 'yyyy/MM/dd HH24:mi:ss') AS now FROM dual
  </select>
</mapper>
```