KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

스프링과 MySQL Database 연동 (심화1)

[KB] IT's Your Life

KB 국민은행

스프링과 MySQL Database 연동

- MyBatis와 관련된 의존 라이브러리를 추가하세요.

스프링과 MySQL Database 연동

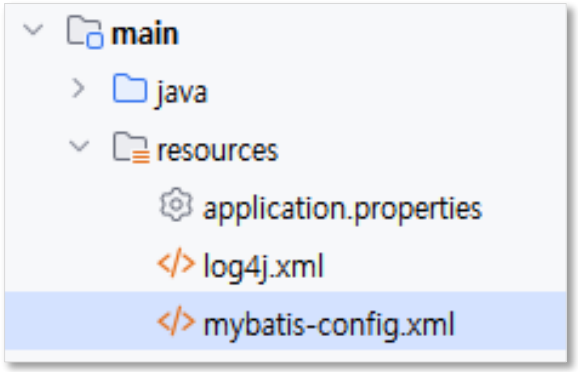
build.gradle

```
                    ...
// 데이터베이스
implementation 'com.mysql:mysql-connector-j:8.1.0'
implementation 'com.zaxxer:HikariCP:2.7.4'

implementation "org.springframework:spring-tx:${springVersion}"
implementation "org.springframework:spring-jdbc:${springVersion}"

implementation 'org.mybatis:mybatis:3.4.6'
implementation 'org.mybatis:mybatis-spring:1.3.2'
                    ...
```

→ gradle sync

스프링과 MySQL Database 연동

- 다음 처럼 mybatis 설정 파일을 만들고 기본 설정 구조를 추가하세요.



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>
```

스프링과 MySQL Database 연동

- **RootConfig에 MyBatis를 위한 기본 설정을 하세요.**
  - SqlSessionFactory 빈 등록
  - DataSourceTransactionManager 빈 등록

스프링과 MySQL Database 연동

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:/mybati
s-config.xml"));
    sqlSessionFactory.setDataSource(dataSource());
    return (SqlSessionFactory) sqlSessionFactory.getObject();
  }

        @Bean
        public DataSourceTransactionManager transactionManager(){
                DataSourceTransactionManager manager = new DataSourceTransactionManager(dataSource());
                return manager;
        }
}
```

스프링과 MySQL Database 연동

- RootConfigTest에서 SqlSessionFactory가 빈등로 정상 등록되었는지 확인하세요.

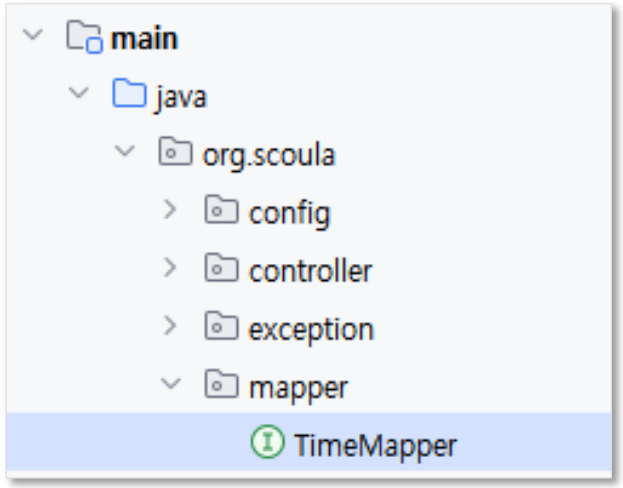
test:: RootConfigTest.java

```
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());
        }
    }
}
```

```
INFO : org.scoula.config.RootConfigTest - org.apache.ibatis.session.defaults.DefaultSqlSession@6f6621e3
INFO : org.scoula.config.RootConfigTest - HikariProxyConnection@967643830 wrapping com.mysql.cj.jdbc.ConnectionImpl@
4eb45fec
```

다음처럼 TimeMapper 인터페이스를 추가하세요.

- org.scoula.mapper 패키지
  - TimeMapper 인터페이스 추가



- String getTime() 메서드에 현재시간을 리턴하는 쿼리를 연결하세요.

mapper.TimeMapper.java

```
package org.scoula.mapper;

import org.apache.ibatis.annotations.Select;

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

}
```

스프링과 MySQL Database 연동

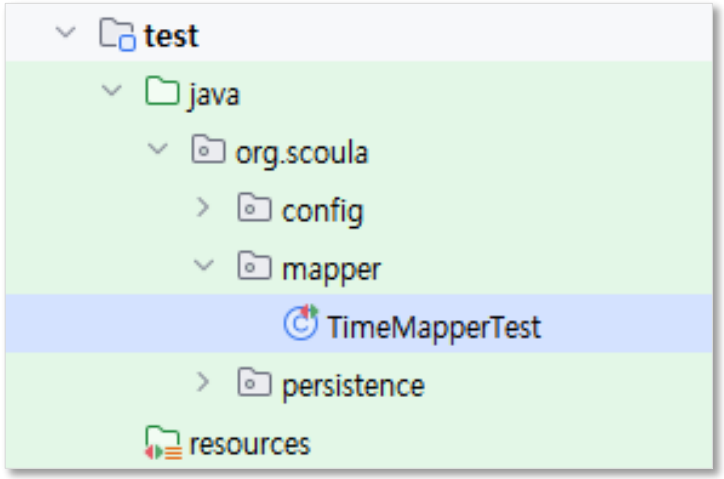
- RootConfig에 Mapper를 스캔하는 설정을 추가하세요.

RootConfig.java – Mapper 설정

```
…
import org.mybatis.spring.annotation.MapperScan;
…

@Configuration
@PropertySource({"classpath:/application.properties"})
@MapperScan(basePackages  = {"org.scoula.mapper"})
public class RootConfig {

        …

}
```

스프링과 MySQL Database 연동

- 다음처럼 단위테스트 TimeMapperTest 를 생성하고, 앞에서 작성한 getTime() 메서드를 테스트하세요.



test :: TimeMapperTest.java

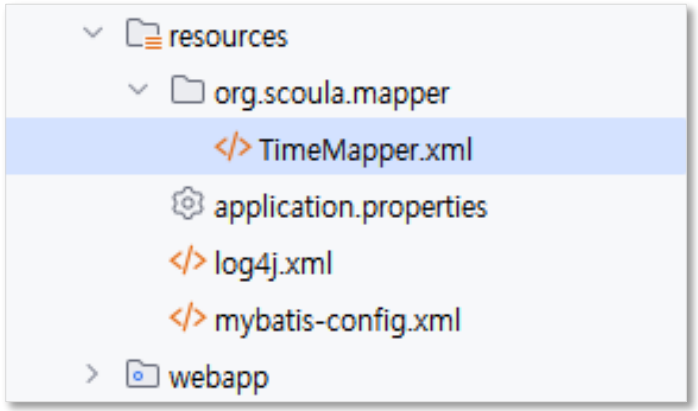
```
...

@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());
        }
}

INFO : org.scoula.mapper.TimeMapperTest - jdk.proxy3.$Proxy32
INFO : org.scoula.mapper.TimeMapperTest - 2023-11-01 17:20:49
```

스프링과 MySQL Database 연동

- **다음 처럼 TimeMapper를 수정하세요.**

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

    public String getTime2();
}
```

- **다음과 같이 TimeMapper.xml을 추가하고 기본 골격을 만들고 getTime2에 해당하는 select 태그를 완성하세요.**



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">

        <select id="getTime2" resultType="string">
                SELECT sysdate()
        </select>
</mapper>
```

String getTime2()의 메서드명

String getTime2()의 리턴타입

스프링과 MySQL Database 연동

- TimeMapperTest에서 getTime2()를 테스트하세요.

test::TimeMapperTest.java

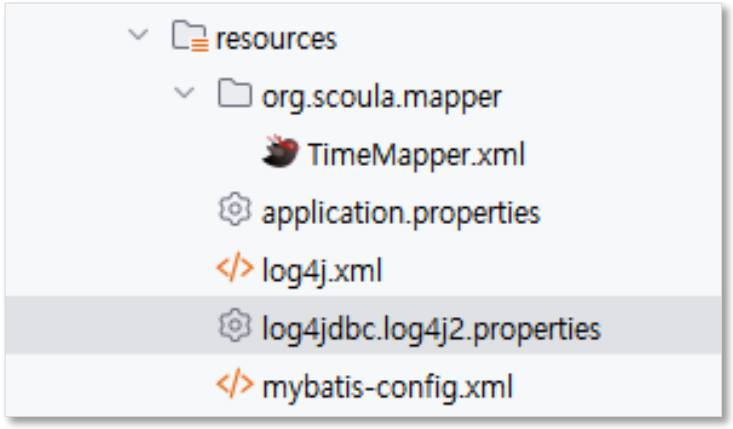
```
package org.scoula.persistence;
…
public class TimeMapperTest {
        …

        @Test
  @DisplayName("TimeMapper의 getTime2()")
        public void getTime2() {

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

        }
}
```

```
INFO : org.scoula.mapper.TimeMapperTest - getTime2
INFO : org.scoula.mapper.TimeMapperTest - 2023-11-01 17:21:58
```

스프링과 MySQL Database 연동

- log4jdbc-log4j2 의존성을 프로젝트에 추가하세요.

- resources/log4jdbc.log4j2.properties에 log4jdbc-log4j2 설정을 추가하세요.



- application.properties에서 DataSource 연결을 위한 드라이버 클래스명과 접속 url을 수정하세요.

build.gradle

```
// Log4j2
implementation 'org.apache.logging.log4j:log4j-api:2.18.0'
implementation 'org.apache.logging.log4j:log4j-core:2.18.0'
implementation 'org.apache.logging.log4j:log4j-slf4j-impl:2.18.0‘

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

스프링과 MySQL Database 연동

log4jdbc-log4j2설정

- 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

```
#jdbc.driver=com.mysql.cj.jdbc.Driver
#jdbc.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
```

log4j2.xml을 다음과 같이 수정하세요.

- 기본 로그 레벨: info
- 단위테스트용
  - 다음 패키지인 경우 warn으로 설정
    - jdbc.audit
    - jdbc.resultset
    - jdbc.connection
  - 다음 패키지인 경우 info로 설정
    - jdbc.sqlonly

- main 운영용
  - 다음 패키지인 경우 warn으로 설정
    - jdbc
  - 다음 패키지인 경우 info로 설정
    - jdbc.sqlonly

test :: resources/log4j2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration>
…
<!-- Logger 설정 -->
<Loggers>
<Root level="INFO">
        <AppenderRef ref="console"/>
</Root>
<Logger name="org.scoula" level="INFO" additivity="false">
        <AppenderRef ref="console"/>
</Logger>
```

test :: resources/log4j2.xml

```
<Logger name="org.springframework" level="INFO" additivity="false">
            <AppenderRef ref="console"/>
</Logger>
<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>
```

KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

스프링과 MySQL Database 연동 (심화2)

[KB] IT's Your Life

KB 국민은행

스프링과 MySQL Database 연동

- 이번 프로젝트를 복사하여 MyBatisSpringLegacy를 만들고 템플릿으로 등록하기위한 기본 파일 구조를 수정하세요.

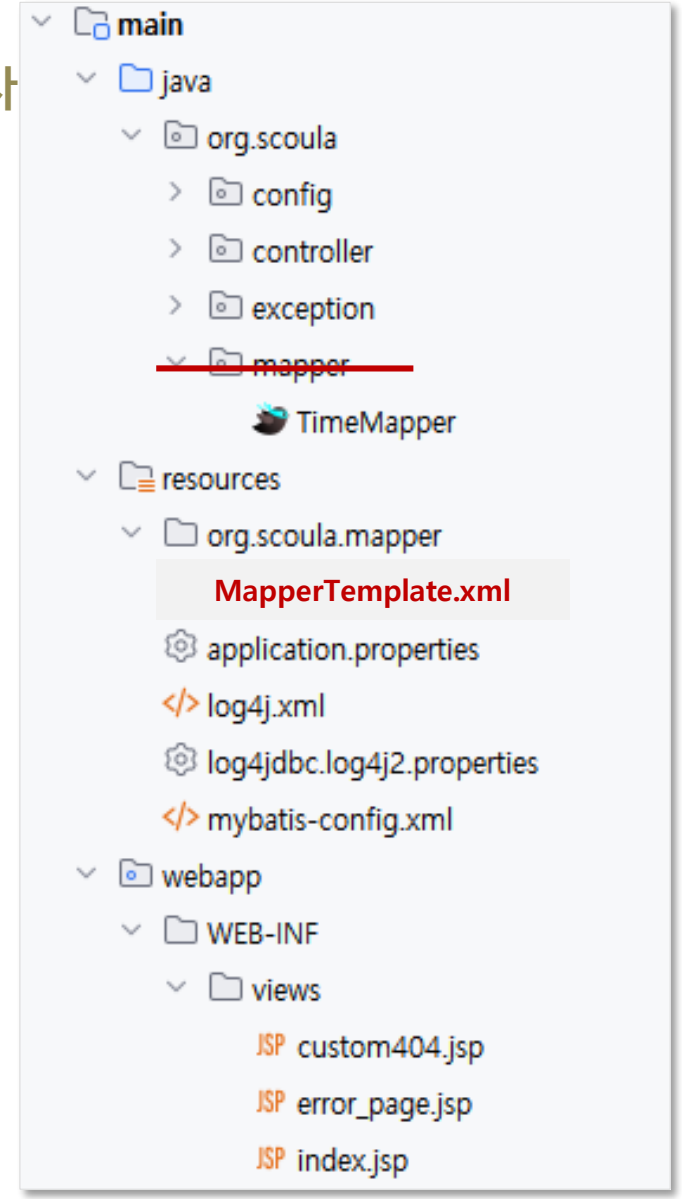
- 완성된 템플릿을 프로젝트 템플릿 MyBatisSpringLegacy로 등록하세요.

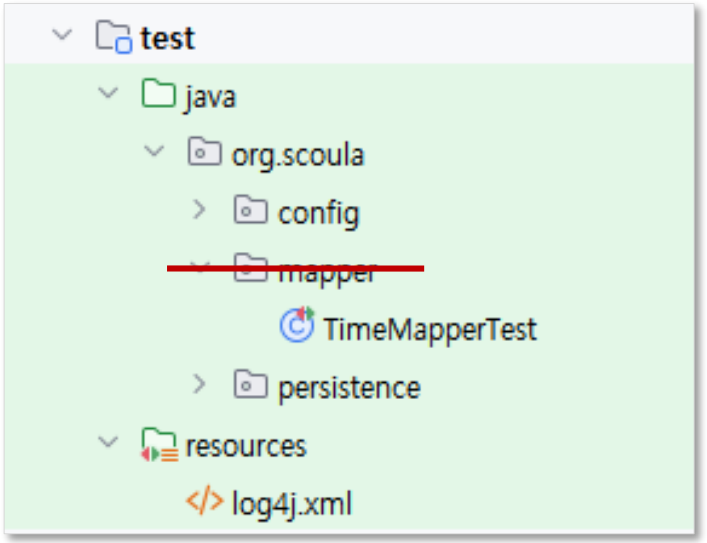
스프링과 MySQL Database 연동

프로젝트 템플릿 만들기

- ex04 → MyBatisSpringLegacy로 복사

- 파일 정리
  - mapper 관련 패키지/파일 삭제
  - mapper 관련 테스트 파일 삭제
  - TimeMapper.xml
    → MapperTemplate.xml





settings.gradle

```
rootProject.name = "MyBatisSpringLegacy"
```

RootConfig.java

```
@Configuration
@PropertySource({"classpath:/application.properties"})
// @MapperScan(basePackages  = {})
public class RootConfig {
                ...
}
```

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...