

간편 만들기

1.pom.xml

원래 있던 pom , 설명.

```
<properties>
<java-version>1.8</java-version>
    <org.springframework-version>4.3.9.RELEASE</org.springframework-version>
    <org.aspectj-version>1.6.10</org.aspectj-version>
    <org.slf4j-version>1.6.6</org.slf4j-version>
</properties>
<!-- DB_POM_1 -->
    <!--오라클 repositories -->
<!--
    <repositories>
        <repository>
            <id>oracle</id>
            <url>http://maven.jahia.org/maven2</url>
        </repository>
    </repositories> -->
<dependencies>

<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-test</artifactId>
    <version>${org.springframework-version}</version>
</dependency>

    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-jdbc</artifactId>
        <version>${org.springframework-version}</version>
    </dependency>

    <dependency>
        <groupId>org.mybatis</groupId>
        <artifactId>mybatis</artifactId>
        <version>3.4.1</version>
    </dependency>

    <dependency>
        <groupId>org.mybatis</groupId>
        <artifactId>mybatis-spring</artifactId>
        <version>1.3.0</version>
```

```

</dependency>

<dependency>
    <groupId>org.bgee.log4jdbc-log4j2</groupId>
    <artifactId>log4jdbc-log4j2-jdbc4</artifactId>
    <version>1.16</version>
</dependency>

<!-- DB_POM_2 -->
    <!-- 오라클 JDBC 드라이버 -->
<!--
    <dependency>
        <groupId>com.oracle</groupId>
        <artifactId>ojdbc6</artifactId>
        <version>12.1.0.2</version>
    </dependency>
-->

    <!-- PostgreSQL 9.4 -->
<!--
    <dependency>
        <groupId>postgresql</groupId>
        <artifactId>postgresql</artifactId>
        <version>9.0-801.jdbc4</version>
    </dependency> -->

    <!-- Mysql 8.0 -->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.13</version>
    </dependency>

```

2. root-context.xml 수정.

resources 추가 파일!!

1)

Namespaces 에서 jdbc,context,mybatis추가.

2)

<!-- Root Context: defines shared resources visible to all other web components -->

<!-- properties -->

<bean

class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">

<property name="locations" value="classpath:/jdbc.properties" />

<property name="fileEncoding" value="UTF-8" />

</bean>

<!-- JDBC-PostgreSQL -->

<bean id="dataSource"

class="org.springframework.jdbc.datasource.DriverManagerDataSource">

<property name="driverClassName"

value="\${jdbc.driverClassName}" />

<property name="url" value="\${jdbc.url}" />

<property name="username" value="\${jdbc.username}" />

<property name="password" value="\${jdbc.password}" />

</bean>

<bean id="sqlSessionFactory"

class="org.mybatis.spring.SqlSessionFactoryBean">

<property name="dataSource" ref="dataSource" />

<property name="configLocation"

value="classpath:/mybatis-config.xml">

</property>

<property name="mapperLocations"

value="classpath:mappers/**/*.Mapper.xml"></property>

</bean>

<bean id="sqlSession" class="org.mybatis.spring.SqlSessionTemplate"

destroy-method="clearCache">

<constructor-arg name="sqlSessionFactory"

ref="sqlSessionFactory"></constructor-arg>

</bean>

<context:component-scan base-package="org.woo.web" />

3.test. (JUnit 4추가.)

:project 우클릭 > properties > Java Build Path > Libraries > add Library > Junit > add
src\test\java\org\woo
DBTest ,DSTest 생성.

```

@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = {
    "file:src/main/webapp/WEB-INF/spring/root-context.xml" })
public class DBTest {

    //    @Test
    //    public void DBConnectionTest() throws Exception {
    //        Class.forName("oracle.jdbc.driver.OracleDriver");
    //        Connection con = DriverManager.getConnection(
    //            "jdbc:oracle:thin:@192.168.0.33:1521:myoracle",
    //            "ora_user",
    //            "woo");
    //        System.out.println(con);
    //        con.close();
    //    }

    //    @Test
    //    public void DBConnectionTest() throws Exception {
    //        Class.forName("org.postgresql.Driver");
    //        Connection con = DriverManager.getConnection(
    //            "jdbc:postgresql://localhost:5432/postgres",
    //            "postgres",
    //            "1234");
    //        System.out.println(con);
    //        con.close();
    //    }

    @Test
    public void DBConnectionTest() throws Exception {
        //mysql driver name 5version
        //com.mysql.jdbc.Driver
        //8버전 부터 driver name이 변경 되었고 Timezone을 설정하도록 변경 되었다.
        (& == & 랑 같고 xml에서는 &을 읽지 못한다.)
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection(

            "jdbc:mysql://localhost:3306/foodmap?useSSL=false&serverTimezone=Asia/Seoul",
            "woo",
            "1234");
        System.out.println(con);
        con.close();
    }
}

```

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = {
    "file:src/main/webapp/WEB-INF/spring/root-context.xml" })
public class DSTest {
```

```
    @Inject
    private DataSource ds;
```

```
    @Test
    public void dsTest()throws Exception{
        Connection con = ds.getConnection();
        System.out.println(con);
        con.close();
```

```
    }
```

```
}
```

-----SqlTest