Spring - MyBatis 연동

솔직히 토씨 하나 틀리지않고 처음부터 끝까지 따라했었다면 쉽게 연동을 하고 지나칠 수 있는 부분이였겠지만, 여러가지 실수라고 봐야하나 싶은 부분이 생겼고, 그 결과로 여러 오류를 몸소 느껴보면서 진행하게 되었습니다.
Spring과 MyBatis의 연동이지만, 생각보다 간단하진 않더라구요..
검색도 잘 못하는거 같기도 해요 전...생각해보면 어쨋든,

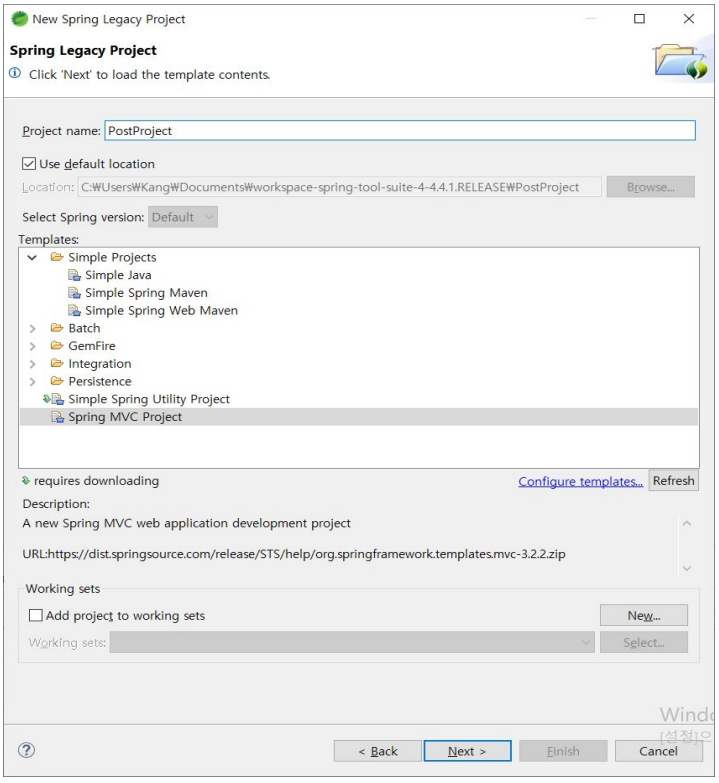
- 난 연동에서 단 한번도 실패해 본 적이 없다.
- 난 MyBatis의 동작에 대해 완벽히 이해하고있다.
- 여기서 왜 오류가 떠서 이러고있는건지 이해가 안간다.

싶으신 분들은 딱히 안보셔도 될거같습니다.

사실, 뒤죽박죽이 되어버린 과정때문에 과연 무엇이 문제여서 이전에 되지않았는가? 에대한 질문에는 답할 수 없게 되었습니다.
버전도 바꿔보고, 모듈도 새로 받아보고, 필요없는게 있어서 그런가 싶은 마음에 프로젝트도 새로 만들어보고.. 여러가지 방식을 시도해봤으며,
일단, 그 이전에 만들었던 프로젝트들은 한두개 바꾼다고 실행되거나 하진 않는거 같습니다.
어쨋든, 완벽하진않치만, 시작해 보겠습니다.
뭐 어때요, 우린 초보인걸요.

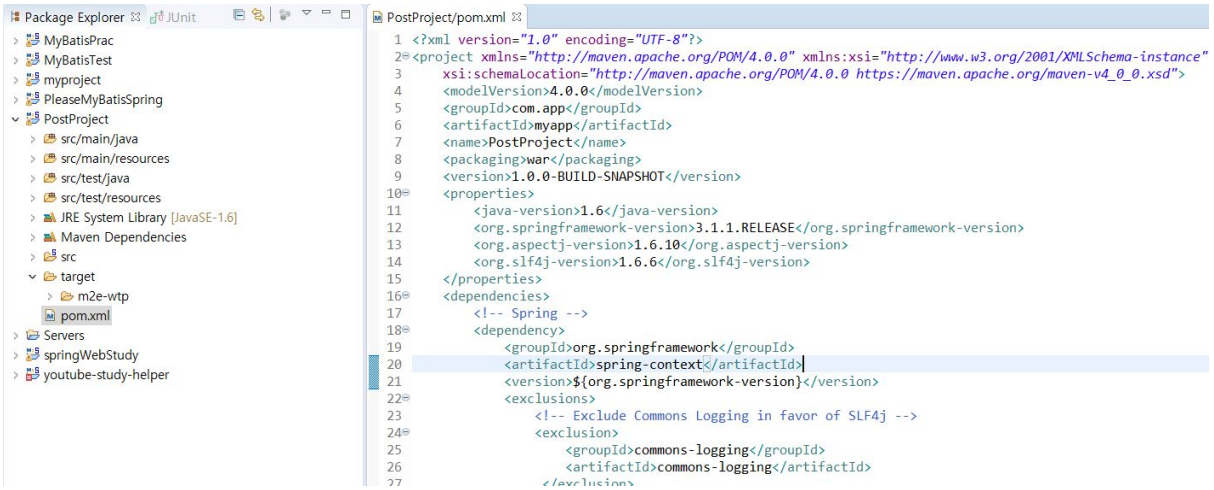
최종적인 연동에는 성공했지만, 다시 보여주기 위해서 처음부터 프로젝트를 생성하고 진행하는 과정을 가져보려고 합니다.

1. STS를 이용한 프로젝트의 생성.



당연히 프로젝트를 먼저 생성해야 합니다.
저의 경우엔 STS를 설치한 후 최대한 Spring의 이론을 공부하면서 보아왔던 환경과 가장 비슷한 환경을 찾으려고 했었고, 그 결과 Spring Legacy Project를 사용하게 되었습니다.
현재의 STS4에서는 바로 사용 할 수는 없으며,
Help - Eclipse marketplace
이곳으로 들어가 STS3에 관련된 모듈을 다운받아 주셔야 정상적으로 Spring Legacy Project를 생성하실 수 있습니다.

Spring Legacy Project를 생성 할 수 있게된다면 위 사진처럼 Spring MVC Project를 선택하고 프로젝트를 생성합니다.
그 뒤에 패키지 명을 지정하라고 나오는데 이부분은 편한대로 작성해 주시면 됩니다.
예제에서는 com.app.myapp을 사용합니다.
그러면, 이제 프로젝트가 생성되었습니다.



pom.xml을 보시게 되면 이런 화면을 보실 수 있습니다.
이제 저희는 여기서 아래의 작업을 하려고 합니다.

1. java버전 수정.
2. 프레임워크 버전 수정.
3. dependency추가.
4. dependency버전 수정.
    + 이 과정에 다시 할때도 필요하다면 maven project의 update와 install.
    + 아마 하게 되겠죠?
5. root-context.xml 작성.
6. junit을 통한 단계별 datasource, mybatis, sqlsession과 DAO 테스트
7. 성공하길 바라며 기도.

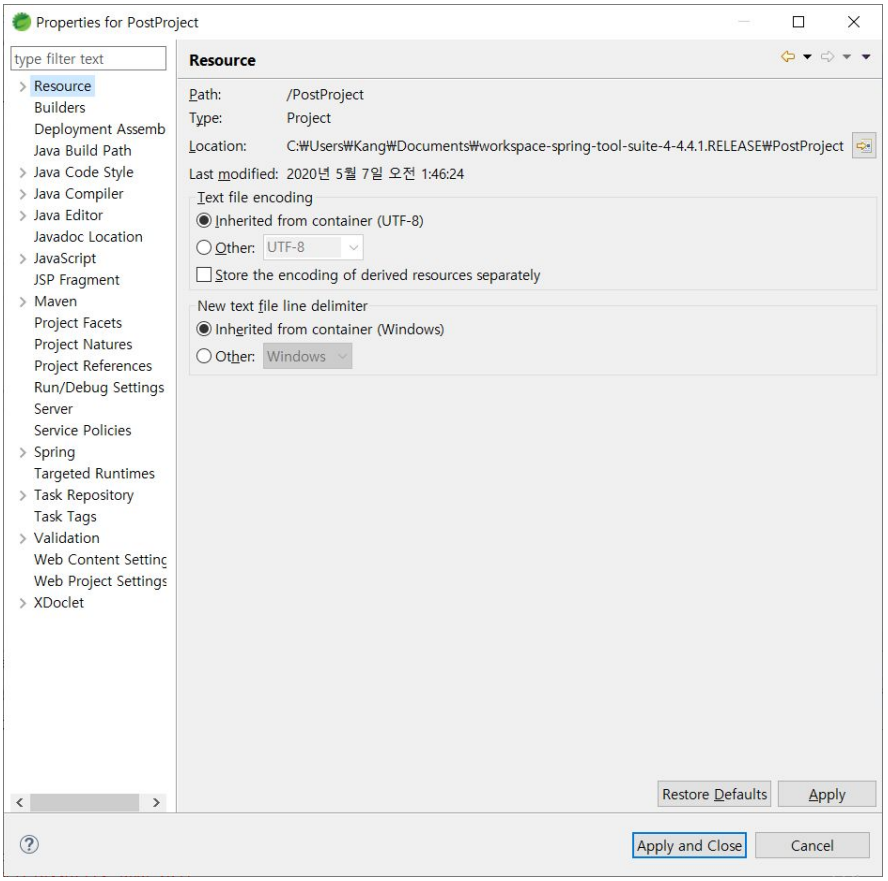
이렇게 입니다. 참 쉽죠? 쉬워야 할텐데요...(솔직히 다시하면 성공할지 저도 모르겠어여..)

혹시 이 글을 보고 이렇게 까지 해야할 필요가 있나? 라고 생각하실 수 있는데
저는 이렇게 해서 성공했고, 아마 이렇게 해야되지 않을까? 라는 생각을 가지고 작성중에 있습니다. 편한방법이 있다면, 당연히 그 방법이 맞습니다.

참고로 Junit을 굳이 왜 해야하냐 라고 물어보시는분들은 안하셔도 좋은데
뒷감당은 알아서... 저도 그렇게 생각하는 사람중 하나였는데
이번에 12시간동안 아무런 스택트레이스 없는 404페이지를 보고있었더니 죽겠네요 네..

1) Java 버전 수정

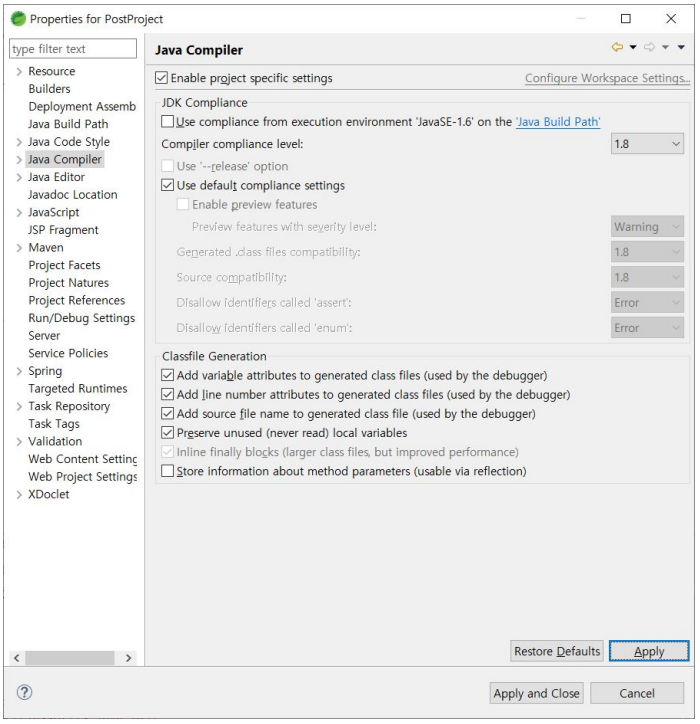
프로젝트를 우클릭하여 Properties를 눌러줍니다.

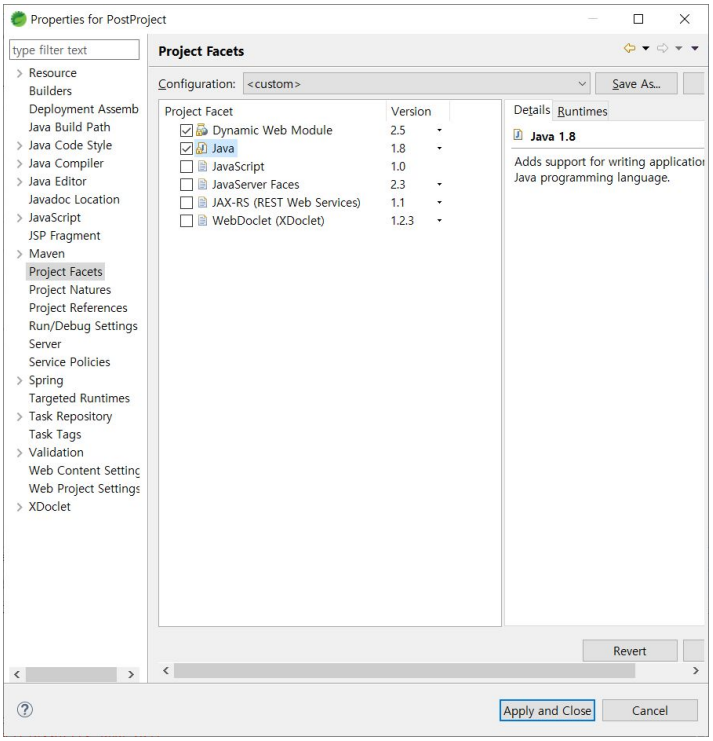


이런 창을 보실 수 있습니다.

프로젝트 혹은 STS자체 설정을 통해 미리 캐릭터셋을 UTF-8로 설정해주시면

프로젝트를 진행하는데 편하실 수 있습니다. 이 부분은 따로 검색을 진행해 주세요.





위의 두 사진과 같이 Java Compiler란에 들어가서 버전을 1.8을 지정해줍니다.
이때, 위와같이 제공되는 1.8을 사용해도되고, 따로 다운받은 1.8버전의 JDK가 있다면, 그걸 사용하셔도 상관없습니다.
JavaCompiler에 버전을 변경시켜주면 프로젝트에 오류가 발생하는것을 확인 할 수 있습니다.
어디 파일에 오류가 있다라고 알려주지는 않지만, 프로젝트아이콘에 빨간색 X자가 선명합니다.
위의 두번째사진, Project Factes를 들어가 줍니다.
여기서 Java의 버전을 1.8로 설정해주시면 됩니다.

다시 pom.xml로 돌아와서,

```
<version>1.0.0-BUILD-SNAPSHOT</version>
<properties>
    <java-version>1.8</java-version>
    <org.springframework-version>3.1.1.RELEASE</org.springframework-version>
    <org.aspectj-version>1.6.10</org.aspectj-version>
    <org.slf4j-version>1.6.6</org.slf4j-version>
</properties>
     </plugin>
     <plugin>
         <groupId>org.apache.maven.plugins</groupId>
         <artifactId>maven-compiler-plugin</artifactId>
         <version>2.5.1</version>
         <configuration>
             <source>1.8</source>
             <target>1.8</target>
             <compilerArgument>-Xlint:all</compilerArgument>
             <showWarnings>true</showWarnings>
```

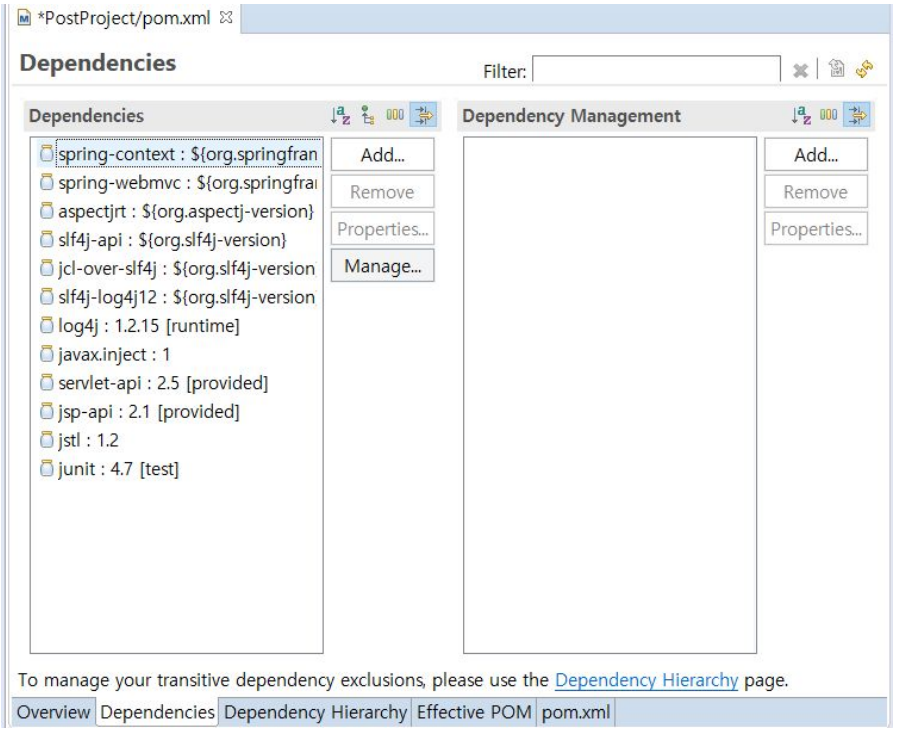
이 두 부분에 1.6 을 1.8로 수정시켜 줍니다.
그럼 일단 자바 버전 수정은 끝났습니다. 제기억이 맞다면요.

2) SpringFramwork버전 수정.

```
<version>1.0.0-BUILD-SNAPSHOT</version>
<properties>
    <java-version>1.8</java-version>
    <org.springframework-version>4.3.2.RELEASE</org.springframework-version>
    <org.aspectj-version>1.6.10</org.aspectj-version>
    <org.slf4j-version>1.6.6</org.slf4j-version>
</properties>
```

방금 Java버전을 수정했던 곳 바로 아래에 springFramwork 버전은 4.3.2.RELEASE로 수정해줍니다.
이 부분은 다른 Dependency에서 ${org.springframwork-version}으로 참조하게 됩니다.
이제, Dependency를 추가해줍니다.

3) Dependency 추가



pom.xml에서 하단에 Dependencies탭을 선택하게되면 위와같은 창이 보이게되며,
Add버튼을 통해 편리하게 Dependency를 다운받아 사용할 수 있습니다.
maven특유의 의존전이역시 감사할 따름인데
사용하고자 하는 jar파일역시 이를통해 쉽게 찾을수찾아 다운받을 수 있습니다.

저희의 목표는 스프링과 mybatis의 연동이기때문에 아래와 같은 dependency를 추가시켜 줍니다.

mybatis
mybatis-spring
spring-jdbc
spring-test
mysql-connector-java

```
<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.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>${org.springframework-version}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-test</artifactId>
    <version>${org.springframework-version}</version>
</dependency>
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.36</version>
</dependency>
```

이와 같이 지정해 주시면 되겠습니다.
이대로 끝나면 좋겠지만, 나중에 Junit Test시에 오류로 인해 Junit 버전역시 바꿔주셔야 합니다.

```
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
</dependency>
```

아, 그리고 이유는 모르겠는데 <scope>test<scope>를 삭제해 주세요.
Junit으로 테스트시에 어노테이션이 인식이 안됩니다.

아직 한발 남았습니다.
servlet버전을 바꿔주셔야 합니다.

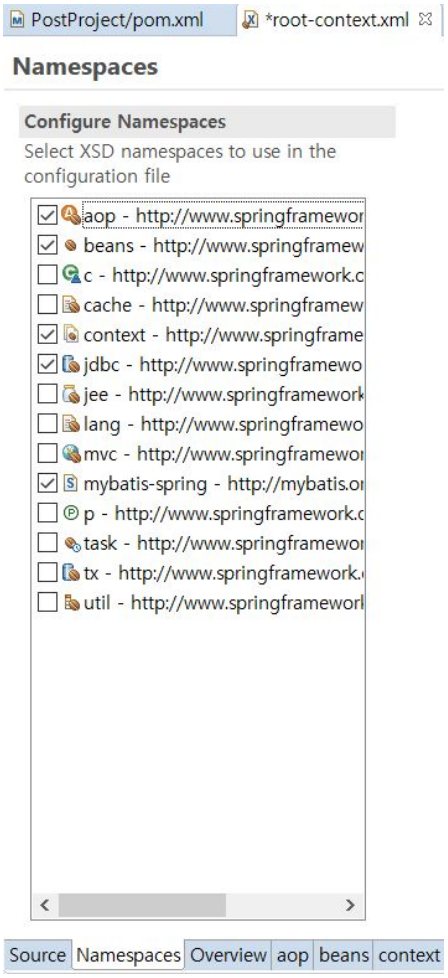
```
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>3.1.0</version>
    <scope>provided</scope>
</dependency>
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>3.1</version>
    <configuration>
        <source>1.8</source>
        <target>1.8</target>
```

기존에 javax.servlet 그룹의 servlet-api 2.5버전을

javax.servlet의 javax.servlet-api 3.1.0으로 바꿔주셔야합니다.
간혹 오류가 뜰경우 artifactId에 javax를 안붙였거나, 버전에 마지막에 3.1.0을 3.1이라고 적으셨을 수도 있습니다.
마지막으로, 제일아래, Java버전 변경시에 수정했던 곳의 maven-compiler-plugin의 버전을 3.1로 수정하고 저장을 합니다.
저장을 하게되면 maven이 자동으로 지금까지 추가시켰던 의존에 대하여 다운및 추가시켜줍니다.

사실 이부분 하면서 저장누르면 분명히 어딘가 안되겠지란 생각이 가득했었는데 그렇기때문에 maven update, project clean, maven install 등등 여러가지 했었는데 바로 되는걸 보면 필요없는 과정이 아니였나, 라는 생각을 잠깐 해보게 됩니다.

아, 한가지 더 남았습니다. root-context.xml로 이동해줍니다.



하단의 Namespaces탭을 선택합니다.
aop, beans, context, jdbc, mybatis-spring을 체크해 줍니다.
그 결과 해당하는 네임스페이스가 추가되었음을 확인 할 수 있습니다.

4) root-context

이전의 버전수정파트는 다운받으면서 같이 수정했기때문에 넘어가도록 하겠습니다.
우선, DataSource가 있어야합니다.
MySql과 그 안에 Table이라던가 하는것은 이미 있다고 생각하고 진행하겠습니다.

아래와 같은 소스를 root-context에 추가시켜줍니다.

```
<bean id="datasource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="driverClassName" value="com.mysql.jdbc.Driver" />
    <property name="url" value="jdbc:mysql://localhost:3306/youtubeschema?serverTimezone=UTC" />
    <property name="username" value="root" />
    <property name="password" value="1234" />
</bean>
```

주의할점으로는 두가지가 있습니다.
하나는 mysql의 한국시간 적용문제라고 해야하나요? 그 문제로 인해 url의 value값에 제일 뒤에 serverTimezone=UTC를 쿼리로 함께 적어주셔야 합니다.
또한 두번째로 자바에서는 최근 버전의 mysql을 사용할 때에는 com.mysql.cj.jdbc.Driver를 사용하는것으로 알고있습니다.
mysql의 버전을 현재 스프링에서 5.1.36으로 낮췄기 때문일 수는 있지만, 이전에 사용시에 당연히 com.mysql.cj.jdbc.Driver를 사용했었고 dataSource를 찾을 수 없다는 오류가 발생했었습니다.
버전문제일 가능성이 농후하며, 여기서는 com.mysql.jdbc.Driver를 사용합니다.
이제, DataSource를 빈으로 등록하였습니다. 테스트를 해 봐야 합니다.

위와 같이 src/test/java에 DataSourceTest라는 이름으로 테스트 파일을 작성해줍니다.
실행할때 스프링이 제공하는 JUnit으로 테스트를 하겠다, 설정컨텍스트는 여기있고,
이녀석을 주입받아 사용할꺼고
테스트 전 / 후에는 이걸 한번씩 호출하고
이것에대한 테스트를 진행하고 등등 여러 관련된 내용이 있습니다.
JUnit에 대하여 궁금한점이 있으면 관련된 내용을 찾아보시길 바랍니다.
이와같이 작성 한 후, Run as -- Junit Test를 실행시켜주시면 됩니다.
오류가 발생할 경우 이 문서 제일 하단에 포함된 링크를 열어서 찾아보시길 바랍니다.
도움이 될 수도있습니다.
(특정 오류에 대한 문제 혹은 에러에 대한 내용을 잘 읽어보세요와 같은 글입니다. 왜 에러가 떴는지 에 대한 정보를 얻을 실 수도 있습니다.)
성공하셨다면, 다음으로 넘어갑니다.
다음은 MyBatis와 관련 설정에 대한 내용입니다.

5) SqlSession / MyBatis설정 + Junit테스트

해보니 마지막이 좀 길어지게 됐습니다.
이전에는 다른분의 포스팅을 보면서 3개의 테스트를 어떻게 나눠서 작성했었는데,

이게 여러번 다시 프로젝트를 만들어서 해보고, 다른분들꺼도 참고하면서 하다보니 테스트가 섞인건지.. 약간 이부분은 단위테스트라기보단 섞여있네요..네…
일단 해보겠습니다
가장먼저 root-context.xml입니다

```
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="datasource"></property>
    <property name="configLocation" value="classpath:/mybatis-config.xml"></property>
    <property name="typeAliasesPackage" value="domain"/>
    <property name="mapperLocations" value="classpath:mappers/*.xml"></property>
</bean>

<bean id="sqlSessionTemplate" class="org.mybatis.spring.SqlSessionTemplate">
    <constructor-arg index="0" ref="sqlSessionFactory"/>
</bean>

<context:component-scan base-package="domain" />
<context:component-scan base-package="interfaces" />
<context:component-scan base-package="imple">
    <context:include-filter type="annotation" expression="org.springframework.stereotype.Repository"/>
    <context:include-filter type="annotation" expression="org.springframework.stereotype.Service"/>
</context:component-scan>
```

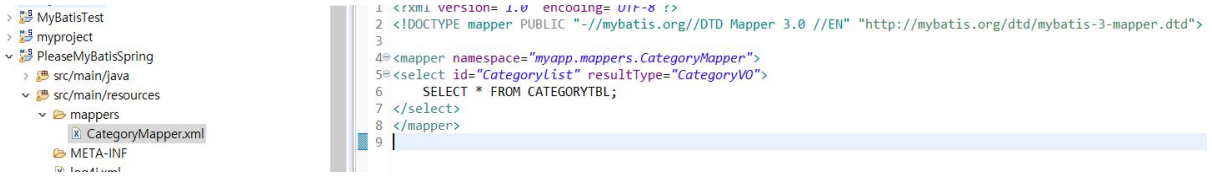
이 코드를 추가시켜주셔야 합니다.
DataSource정보, MyBatis설정정보 그리고 MyBatis를 통해 실행시킬 행위가 적힌 Mapper파일에 대하여 연결을 시켜주는 sqlSessionFactory를 빈으로 등록시킵니다.
sqlSessionTemplate빈을 통해 MyBatis를 통한 쿼리문의 실행이 가능하게 됩니다.

아래부분의 context:component-scan에 관해서는 원래는 component-scan의 경우에 @Controller, @Repository, @Service등에 대한 특정 몇몇 어노테이션들에 대하여 해당 어노테이션이 선언되어있을경우 자동으로 해당 패키지 하위에 대하여 빈 등록을 해주는 어노테이션으로 알고있습니다.
그런데 오류를 여러번 겪으면서 좀 더 확실히 해주기 위해서 include-filter를 사용한 모습입니다.
여기서 주의할점이 있습니다. 아직도 이유는 모르겠습니다.
servlet-context를 보시면 아래와 같습니다.

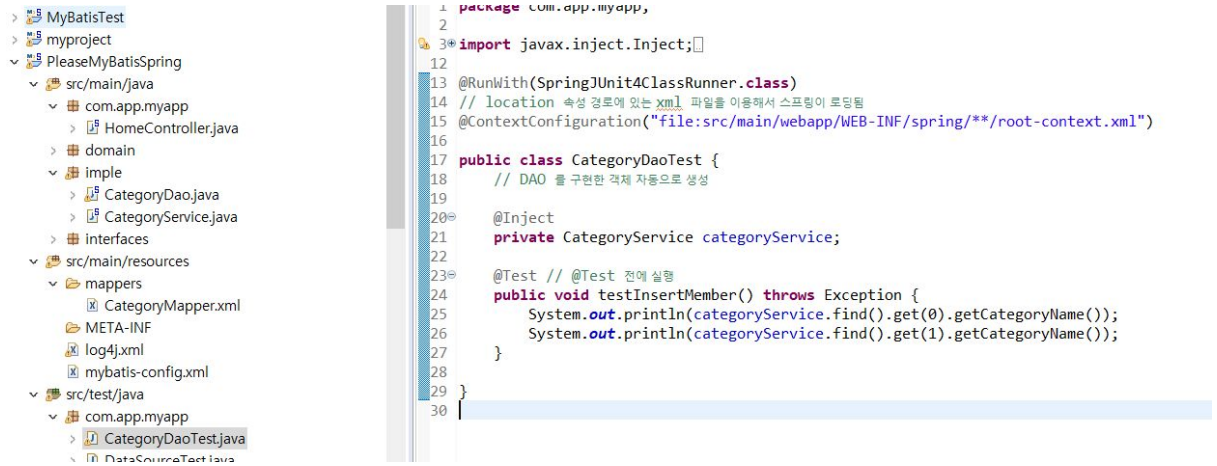
```
    <!-- Resolves views selected for rendering by @Controllers to .jsp resources in the /WEB-INF/views directory -->
    <beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <beans:property name="prefix" value="/WEB-INF/views/" />
        <beans:property name="suffix" value=".jsp" />
    </beans:bean>

    <context:component-scan base-package="com.app.myapp" />
```

많이 본 모습이 눈에 띄입니다.
com.app.myapp은 src/main/java에 기본으로 등록되어있는 패키지입니다.
그렇기 때문에 처음에 저는 이곳에 component-scan을 모두 적으면 되겠구나, 라는생각을 했었습니다.
어차피 컨트롤러 빈이나 서비스등과 같은 녀석들에 대한 빈 등록이나 다똑같다고 생각을 했는데…
참 신기합니다. 위에 녀석들은 이곳에서 빈 등록할경우 빈등록이 되지않습니다.
혹시나해서 root-context,xml로 옮겼을때 실행이 되버려서 깜짝놀랐었는데.
이유를 아신다면 알려주세요…
일단, 다음으로 넘어가겠습니다.

src/main/resources폴더 하위에 mybatis-config.xml파일을 생성합니다.
위와같이 MyBatis configuration에 대한 DTD임을 명시해 줍니다.
<configuration></configuration>만 있어도 상관은 없습니만,
카멜케이스로 변경해주는 기능을 사용해 봅시다.
configuration하단에 setting구성요소를 사용합니다.
mapUnderscoreToCamelCase에 대하여 값을 true로 지정해줍니다.
이 기능같은경우에는 보통은 낙타표기법으로 작성하는 자바 변수명과 언더바표기법을
사용하는 SQL의 컬럼간에 매핑을 시켜주는 방식입니다.
사용시에 변수명과 컬럼명이 같다면 사용하실 필요가 없으며, 아예 형식문제가 아니라
다른뜻을 가지고있다면 사용해도 의미가 없는것으로 알고있습니다.
언더바를 제외하고 같은 이름을 가졌을때에 서로 매핑시켜 주는 역할을 합니다.

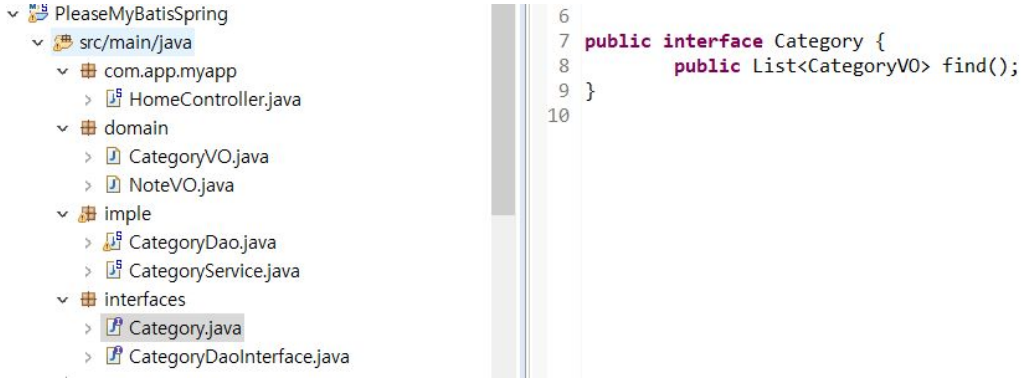


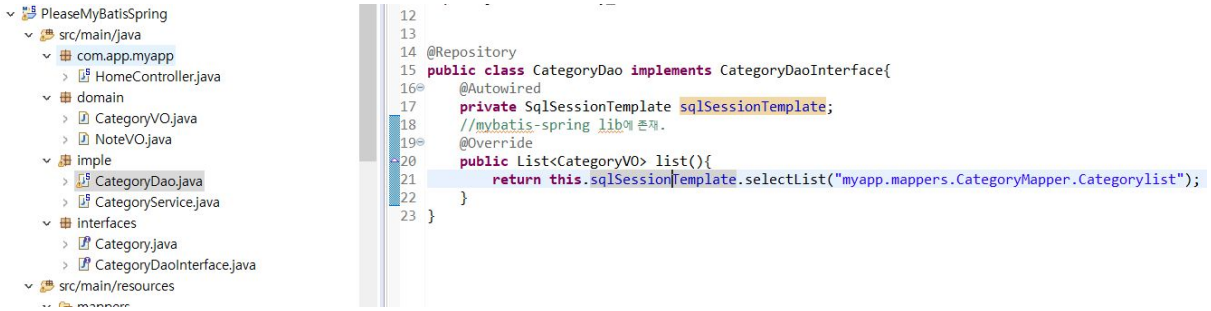
mappers라는 폴더를 하나 src/main/resources폴더 하위에 만들고 Mapper파일을 하나
작성합니다.
저의 경우에는 CategoryMapper.xml이란 이름을 사용하며, 여기서 부터는 실제 여러분이
사용하실 내용과 제가 사용한 내용이 다를 수 있습니다.
저는 DB에 카테고리라고 저장해놓은 목록을 뽑아볼 예정이며,
여러분은 뭐 학생의 이름을 뽑겠다! 일 수도 있으니까요. 상황에 맞게 적어주시면 됩니다.
xml에 mapper파일이라는 DTD를 선언해 줍니다.
mapper는 namespace를 가지며, 나중에 서비스단에서 네임스페이스 + 쿼리문의 id속성을
가지고 실행해야 할 쿼리문을 찾아가게 됩니다.
보통은 네임스페이스를 패키지명 + 클래스명으로 한다고 했던것 같네요.
저의 경우에는 myapp.mappers.CategoryMapper라는 네임스페이스에 Categorylist라는 id를
부여해 주었습니다.
<select>는 조회에 사용되는 속성이며, insert 혹은 delete등에 대해서는 <select>구성요소가
아닙니다. mybatis관련 게시글을 찾아보시기 바랍니다.
간다하게 설명하면 위에서 resultType의 경우에는 쿼리의 결과로 반환되는 resultSet이 담길
자바 객체를 의미합니다.
이곳에서는 파라미터를 사용하지않아 작성하지 않았지만, 파라미터가 필요할경우
parameterType을 선언하여 파라미터를 지정해 줄 수 있습니다.
VO객체를 선언하여 setter를 통해 조건을 집어넣고, 해당 VO객체를 sqlSessionTemplate의
메서드 실행시에 파라미터로 주입하여 사용 하실 수 있습니다.

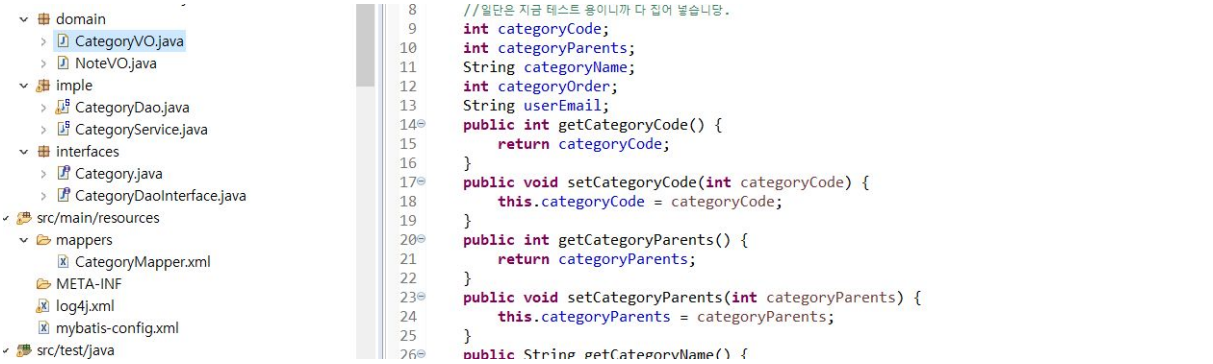


다시, src/test/java폴더 하위에 CategoryDaoTest.java라는 파일을 만들어 줍니다.
위와같은 내용을 가지고있으며, CategoryService를 주입받아 결과를 출력하도록 Test합니다.
CategoryService는 Dao객체를사용하여 선언해놓은 find()메소드를 통해 DB의 CategoryList를 추출합니다.
이때, Dao객체에서는 sqlSessionTemplate에서 자체적으로 선언되어있는 메서드를 이용하여 쿼리문을 호출하여 사용합니다.

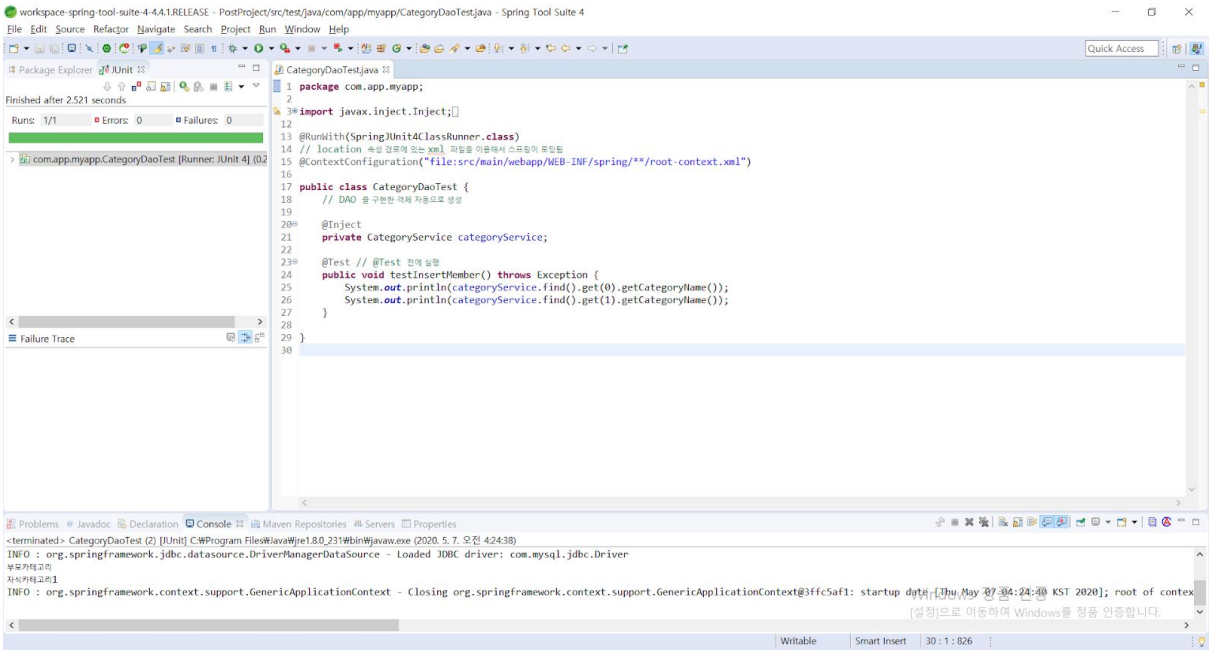
아래는 이 테스트에 사용한 CategoryVO객체, CategoryDao, CategoryDaoInterface, CategoryService, 그 인터페이스인 Category에 대한 소스입니다.







이렇게 보면 진짜 복잡한건 없는데 뭔가 고생을 엄청...
일단 여기서 중요한건 DAO클래스 입니다.
sqlSessionTemplate객체를 자동의존주입받으며 사용하는 부분입니다.
이때, selectList라는 자체적으로 선언된 메서드를 호출하면서 특정 String값을 넘겨줍니다.
어디서 많이 봤던모습입니다.
바로, CategoryMapper.xml에서 선언한 해당 mapper의 네임스페이스와 그 id값 입니다.
이처럼 MyBatis에서는 Mapper와의 연결을 이런식으로 처리하게 됩니다.
물론 이부분이 틀렸을경우에는 당연히 해당 Mapper를 찾지 못하게 되며 오류가 발생합니다.
주의해야 합니다.



뭔가 굉장히 먼 길을 돌고 돌아 돌아온 느낌입니다.
DB에 저장해놓은 부모카테고리, 자식카테고리라는 글이 보입니다.
기나긴 여정의 끝, 감격의 순간입니다.

확실하지 않은 오류도 있고, 왜 안됐었는지, 그리고 지금에 와서 왜 되는지 잘 모르겠는 부분도 분명이 존재합니다만, 이유를 모두 다 알면 좋겠지만 참, 어렵습니다.

아래는 제가 문제를 해결하면서 봤었던 글들입니다.
좀 더 정리가 잘 되어 있을 수 있으며, 참고할만 할 수 있습니다.

요약하자면,

1) 버전문제일 수 있으며,
2) 빈 등록 혹은 빈 중복의 문제가 있을 수 있고
3) 경로 설정이 되지않을경우 등록 자체가 안되어 오류가 발생할 수 있습니다.

-스프링 MyBatis연동
https://hanazuou.tistory.com/160
https://www.hanumoka.net/page/16/
https://pnot.tistory.com/5
https://gabrielyj.tistory.com/149
https://kookyungmin.github.io/server/2018/08/13/spring_07/
https://velog.io/@wimes/2.-%EA%B0%9C%EB%B0%9C%ED%99%98%EA%B2%BD%EC%84%A4%EC%A0%95-Spring-MyBatis-MySQL%EC%9D%98-%EC%84%A4%EC%A0%95-2zk4cf5gof
-Spring] 에러 : Colund not load CacheAwareContextLoaderDelegate
https://m.blog.naver.com/PostView.nhn?blogId=ege1001&logNo=220908021174&proxyReferer=https:%2F%2Fwww.google.com%2F

-clathpass와 관련하여
https://developer-joe.tistory.com/225
-classNotFoundException
https://stackoverflow.com/questions/24259508/java-lang-classnotfoundexception-org-springframework-core-io-resource
-Unsatisfied Error
https://beagle-dev.tistory.com/122
-could not autowired field → 빈 미할당 혹은 중복 주입에 신경써야합니다.
https://okky.kr/article/551253
-injection of resource dependencies failed. :: NoSuchBeanDefinitionException
https://mkil.tistory.com/301
-No found for dependency: expected at least 1 bean which qualifies as autowired …
https://stackoverflow.com/questions/26095881/no-found-for-dependency-expected-at-least-1-bean-which-qualifies-as-autowire-ca
-자바버전변경
https://yongtech.tistory.com/98
https://stackoverflow.com/questions/18463067/eclipse-updating-the-compiler-compliance-to-1-7
-okky 스프링 에러 관련 오류. 에러 스택트레이스 잘 보란 소리
https://okky.kr/article/272069
-Error creating bean with name 'XXX'
https://liante0904.tistory.com/113
-이건 왜 찾아봤었지.. maven install/update 관련할때 찾아본거같은데..포스팅에는없어요
https://stackoverflow.com/questions/19655184/no-compiler-is-provided-in-this-environment-perhaps-you-are-running-on-a-jre-ra
-Junit 테스트 에러. :: 버전을 통일시켜 주세요
https://bigzero37.tistory.com/19
-기타오류
https://pnot.tistory.com/6
-Junit에 관하여
https://countryxide.tistory.com/17
-unsatisfied dependencty관련시 경로에 주의
https://epthffh.tistory.com/entry/%EC%BD%94%EB%93%9C%EB%A1%9C-%EB%B0%B0%EC%9A%B0%EB%8A%94-%EC%8A%A4%ED%94%84%EB%A7%81-1-%ED%94%84%EB%A1%9C%EC%A0%9D%ED%8A%B8-%EC%8B%9C%EC%9E%91-%EA%B8%B0%EB%B3%B8%EC%A0%81%EC%9D%B8%EC%84%A4%EC%A0%95