HOL : Spring and MyBatis

Task1. MySQL의 World Database의 City Table 가져오기
1. MariaDB 설치하기
  1)https://mariadb.org
  2)https://downloads.mariadb.org/ MariaDB 10.4.8 Stable
    -mariadb-10.4.8-winx64.msi

2. MyBatisDemo project 생성
  1)In Package Explorer > right-click > New > Java Project
  2)Project name : MybatisDemo

3. src > right-click > New > Package
  1)Package name : com.example

4. Java Project를 Spring Project로 변환
  1)MybatisDemo Project > right-click > Configure > Convert to Maven Project
  2)Project : /MybatisDemo
  3)Group Id : MybatisDemo
  4)Artifact Id : MybatisDemo
  5)version : 0.0.1-SNAPSHOT
  6)Packaging : jar
  7)Finish

  8)MybatisDemo Project > right-click > Spring > Add Spring Project Nature

  9)pom.xml 파일에 Spring Context Dependency 추가하기
```
    <version>0.0.1-SNAPSHOT</version>
    <dependencies>
      <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>5.2.0.RELEASE</version>
      </dependency>
    </dependencies>
```

  10)pom.xml > right-click > Run As > Maven install
    [INFO] BUILD SUCCESS 확인

5. MyBatis library 검색 및 설치
  1)Maven Repositoy에서 'mybatis'로 검색

```
    <!-- https://mvnrepository.com/artifact/org.mybatis/mybatis -->
    <dependency>
        <groupId>org.mybatis</groupId>
        <artifactId>mybatis</artifactId>
        <version>3.5.3</version>
    </dependency>
```

  2)pom.xml에 등록 및 설치

6. MyBatis-Spring library 검색 및 설치
   1)Maven Repository에서 'mybatis spring'으로 검색

```
<!-- https://mvnrepository.com/artifact/org.mybatis/mybatis-spring -->
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis-spring</artifactId>
    <version>2.0.3</version>
</dependency>
```

   2)pom.xml에 등록 및 설치

7. MariaDB Jdbc Driver library 검색 및 설치
   1)Maven Repository 에서 'mariadb'로 검색하여 MariaDB Java Client를 설치한다.

```
<!-- https://mvnrepository.com/artifact/org.mariadb.jdbc/mariadb-java-client -->
<dependency>
    <groupId>org.mariadb.jdbc</groupId>
    <artifactId>mariadb-java-client</artifactId>
    <version>2.5.1</version>
</dependency>
```

   2)pom.xml에 붙여 넣고 Maven Install 하기

8. Spring JDBC 설치
   1)JdbcTemplate를 사용하기 위해 pom.xml에 다음 dependency를 추가해야 함.

```
<dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-jdbc</artifactId>
   <version>5.2.0.RELEASE</version>
</dependency>
```

   2)pom.xml에 붙여 넣고 Maven Install 하기

9. JUnit Library 설치
   1)http://mvnrepository.com에 접근
   2)jUnit으로 검색
   3)jUnit 4.12 version을 pom.xml에 추가

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

   4)pom.xml > right-click > Run As > Maven Install

10. MybatisDemo/resources folder 생성
  1)MybatisDemo project > right-click > Build Path > Configure Build Path
  2)Source Tab > Add Folder
  3)MybatisDemo click
  4)Create New Folder > Folder name : resources > Finish > OK
  5)MybatisDemo/resources(new) 확인
  6)Apply and Close

11. resources/dbinfo.properties file 생성
  1)resources > right-click > New > File
  2)File name : dbinfo.properties > Finish

```
db.driverClass=org.mariadb.jdbc.Driver
db.url=jdbc:mariadb://localhost:3306/world
db.username=root
db.password=javamariadb
```

12. world database downloads
  1)https://dev.mysql.com/doc/index-other.html
  2)Example Databases > world database > Zip
  3)Unzip
  4)MariaDB login 후 world database 실행

13. 여러 Package 생성
  1)/src/com.example.vo
  2)/src/com.example.service
  3)/src/com.example.dao

14. VO class 작성
  1)/src/com.example.vo.CityVO.java 생성

```
package com.example.vo;

public class CityVO {
   private int id;
   private String name;
   private String countryCode;
   private String district;
   private int population;
   public int getId() {
      return id;
   }
   public void setId(int id) {
      this.id = id;
   }
   public String getName() {
```

```
157             return name;
158         }
159         public void setName(String name) {
160             this.name = name;
161         }
162         public String getCountryCode() {
163             return countryCode;
164         }
165         public void setCountryCode(String countryCode) {
166             this.countryCode = countryCode;
167         }
168         public String getDistrict() {
169             return district;
170         }
171         public void setDistrict(String district) {
172             this.district = district;
173         }
174         public int getPopulation() {
175             return population;
176         }
177         public void setPopulation(int population) {
178             this.population = population;
179         }
180         @Override
181         public String toString() {
182             return String.format("CityInfoVO [id=%s, name=%s, countryCode=%s, district=%s,
                population=%s]", id, name,
183                     countryCode, district, population);
184         }
185     }
186
187
188 15. Mapping file 작성 및 MyBatis 설정
189   1)/resources/SqlMapConfig.xml
190
191     <?xml version="1.0" encoding="UTF-8" ?>
192     <!DOCTYPE configuration
193       PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
194         "http://mybatis.org/dtd/mybatis-3-config.dtd">
195     <configuration>
196       <typeAliases>
197         <typeAlias type="com.example.vo.CityVO" alias="cityVO" />
198       </typeAliases>
199
200     </configuration>
201
202   2)/resources/mybatis-mapper.xml
203
204     <?xml version="1.0" encoding="UTF-8"?>
205     <!DOCTYPE mapper
206       PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
207         "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
```

```
208     <mapper namespace="City">
209
210     </mapper>
211
212
213 16. Dao 객체 생성
214   1)mybatis-mapper.xml
215
216     <resultMap id="cityResult" type="cityVO">
217       <result property="id" column="ID" />
218       <result property="name" column="Name" />
219       <result property="district" column="District" />
220       <result property="countryCode" column="CountryCode" />
221       <result property="population" column="Population" />
222     </resultMap>
223
224     <select id="selectCityByName" parameterType="String" resultType="cityVO" resultMap="cityResult">
225       SELECT * FROM world.city WHERE  name = #{name}
226     </select>
227
228   2)/src/com.example.dao.CityDao.java
229
230     package com.example.dao;
231
232     import java.util.List;
233     import com.example.vo.CityVO;
234
235     public interface CityDao {
236       List<CityVO> readAll();
237       CityVO read(String name);
238     }
239
240   3)/src/com.example.dao.CityDaoImpl.java
241
242     package com.example.dao;
243
244     import java.util.List;
245
246     import org.apache.ibatis.session.SqlSession;
247     import org.springframework.beans.factory.annotation.Autowired;
248     import org.springframework.stereotype.Repository;
249
250     import com.example.vo.CityVO;
251     @Repository("cityDao")
252     public class CityDaoImpl implements CityDao {
253
254       @Autowired
255         private SqlSession session;
256
257       @Override
258       public List<CityVO> readAll() {
259         return null;
```

```
260         }
261
262         @Override
263         public CityVO read(String name) {
264            CityVO city  = session.selectOne("City.selectCityByName", name);
265            return city;
266         }
267      }
268
269
270 17. Service 객체 생성
271   1)/src/com.example.service.CityService.java
272
273      package com.example.service;
274
275      import java.util.List;
276      import com.example.vo.CityVO;
277
278      public interface CityService {
279         List<CityVO> getCityList();
280         CityVO getCity(String name);
281      }
282
283   2)/src/com.example.service.CityServiceImpl.java
284      package com.example.service;
285
286      import java.util.List;
287
288      import org.springframework.beans.factory.annotation.Autowired;
289      import org.springframework.stereotype.Service;
290
291      import com.example.dao.CityDao;
292      import com.example.vo.CityVO;
293
294      @Service("cityService")
295      public class CityServiceImpl implements CityService {
296
297         @Autowired
298         CityDao cityDao;
299
300         @Override
301         public List<CityVO> getCityList() {
302            return null;
303         }
304
305         @Override
306         public CityVO getCity(String name) {
307            return this.cityDao.read(name);
308         }
309      }
310
311
```

18. Bean Configuration XML 작성
1)/resources > right-click > New > Spring Bean Configuration File
2)File name : beans.xml > Finish
3)Namespace Tab
4)Check context - http://www.springframework.org/schema/context

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
  http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-3.2.xsd">

<!-- mybatis-spring 설정 -->
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
  <property name="dataSource"  ref="dataSource" />
  <property name="configLocation"   value="classpath:SqlMapConfig.xml" />
  <property name="mapperLocations">
    <list>
      <value>classpath:mybatis-mapper.xml</value>
    </list>
  </property>
</bean>

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

<context:property-placeholder location="classpath:dbinfo.properties" />
<bean id="dataSource" class="org.springframework.jdbc.datasource.SimpleDriverDataSource">
  <property name="driverClass" value="${db.driverClass}" />
  <property name="url" value="${db.url}" />
  <property name="username" value="${db.username}" />
  <property name="password" value="${db.password}" />
</bean>
</beans>
```

19. Project의 Bean 등록 및 의존 관계 설정
1)<context:component-scan> tag 사용
2)@Service, @Repository Annotation을 선언한 class들과 @Autowired Annotation을 선언하여 의존관계를 설정한 Class들이 위치한 Package를 Scan하기 위한 설정을 XML에 해주어야 한다.
3)beans.xml에 다음 code 추가한다.

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

20. Spring TestContext Framework 사용하기
1)MvnRepository에서 'spring text'로 검색
2)Spring TestContext Framework에서 5.2.0.RELEASE click

3)다음 코드를 pom.xml에 붙여 넣기

```
<!-- https://mvnrepository.com/artifact/org.springframework/spring-test -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-test</artifactId>
    <version>5.2.0.RELEASE</version>
    <scope>test</scope>
</dependency>
```

4)pom.xml > right-click > Run As > Maven Install
[INFO] BUILD SUCCESS 확인

5)/src/com.example.test package 생성
6)/src/com.example.test > right-click > New > JUnit Test Case
7)Name : MybatisDemoTest > Finish
8)[New JUnit 4 test] 선택
9)Finish

```
import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

import com.example.service.CityService;
import com.example.vo.CityVO;

@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations="classpath:beans.xml")
public class MybatisDemoTest {
    @Autowired
    CityService cityService;

    @Test
    public void test() {
        CityVO city = cityService.getCity("Seoul");
        System.out.println(city);
    }
}
```

10)만일 해당 객체를 찾을 수 없다는 오류가 계속 발생하면
-해당 Project > right-click > Build Path > Libraries tab
-spring-test-5.2.0.RELEASE.jar 선택 후 [Remove] 로 삭제
-[Add External JARs...] Click
-Local M2 Repository(e.g C:₩Users₩bluee₩.m2₩repository₩org₩springframework₩spring-test₩5.2.0.RELEASE)에서 직접 jar를 선택할 것
-[Order and Export] tab에서 spring-test-5.2.0.RELEASE.jar 선택 후 [Up] button을 클릭
-해당 Project/src 바로 아래까지 올리고 [Apply and Close] Click

21. Test
1)right-click > Run As > Junit Test
2)결과 -> Junit View에 초록색 bar
CityInfoVO [id=2331, name=Seoul, countryCode=KOR, district=Seoul, Poplulation=9981619]

22. All City 읽어오기
1)com.example.service.CityServiceImpl.java

```
@Override
public List<CityVO> getCityList() {
   return cityDao.readAll();
}
```

2)com.example.dao.CityDaoImpl.java

```
@Override
public List<CityVO> readAll() {
   List<CityVO> cityList = session.selectList("City.selectList");
   return cityList;
}
```

3)mybatis-mapper.xml

```
<select id="selectList" resultType="cityVO" resultMap="cityResult">
   SELECT * FROM world.city ORDER BY id DESC
</select>
```

4)MybatisDemoTest.java

```
@Autowired
CityService cityService;

@Ignore @Test
public void test() {
   CityVO city = this.cityService.getCity("Seoul");
   System.out.println(city);
}

@Test
public void test1() {
   List<CityVO> list = this.cityService.getCityList();

   for(CityVO vo : list){
      System.out.println(vo.getId());
      System.out.println(vo.getName());
      System.out.println(vo.getDistrict());
      System.out.println(vo.getCountryCode());
      System.out.println(vo.getPopulation());
      System.out.println("------------------------");
   }
}
```

23. Test
    1)right-click > Run As > Junit Test
    2)결과 -> Junit View에 초록색 bar