1. MyBatis 개요
   1)MyBatis(http://www.mybatis.org/mybatis-3)는 Java Object와 SQL 문 사이의 자동 Mapping 기능을 지원하는 ORM Framework이다.
   2)MyBatis는 SQL을 별도의 file로 분리해서 관리하게 해주며, 객체-SQL 사이의 parameter Mapping 작업을 자동으로 해주기 때문에 많은 인기를 얻고 있는 기술
   3)MyBatis는 Hibernate나 JPA(Java Persistence API)처럼 새로운 DB programming paradiam을 익혀야 하는 부담이 없이, 개발자가 익숙한 SQL을 그대로 이용하면서 JDBC code 작성의 불편함도 제거해주고, domain 객체나 VO 객체를 중심으로 개발이 가능하다는 장점이 있다.

2. 특징
   1)쉬운 접근성과 code의 간결함
      -가장 간단한 persistence framework이다.
      -XML 형태로 서술된 JDBC code라고 생각해도 될 만큼 JDBC의 모든 기능을 MyBatis가 대부분 제공한다.
      -복잡한 JDBC code를 걷어내며 깔끔한 source code를 유지할 수 있다.
      -수동적인 parameter 설정과 query 결과에 대한 mapping 구문을 제거할 수 있다.

   2)SQL 문과 programming code의 분리
      -SQL에 변경이 있을 때마다 Java code를 수정하거나 compile 하지 않아도 된다.
      -SQL 작성과 관리 또는 검토를 DBA와 같은 개발자가 아닌 다른 사람에게 맡길 수도 있다.

   3)다양한 programming 언어로 구현가능
      -Java, C#, .NET, Ruby

3. MyBatis와 MyBatis-Spring을 사용한 DB Access Architecture
   -Application Modules
      --Service --> Repository(Mapper)
   -O/R Mapper
      --MyBatis 3
      --MyBatis-Spring
   -JDBC Intefaces
      --JDBC Basic APIs
      --DataSource(Configuration for Connect)
   -JDBC Implementations
      --JDBC Driver
   -Persistence Layer
      --Database

4. MyBatis를 사용하는 Data Access Layer
   1)Presentation Layer
      -Controller
      -MultiActionController

   2)Service Layer
      -Service
      -ServiceImpl

   3)Data Access Layer
      -Dao
      -DaoImpl

   4)MyBatis Framework

-mapper.xml
-jdbc.properties
-sqlMapConfig.xml
-SqlSessionFactory
-SqlSession

5. MyBatis 3의 주요 component
-그림참조
-http://terasolunaorg.github.io/guideline/5.0.0.RELEASE/en/ArchitectureInDetail/DataAccessMyBatis3.html

6. MyBatis 3의 주요 component의 역할
1)MyBatis 설정file(SqlMapconfig.xml)
-Database의 접속 주소 정보나 Mapping file의 경로 등의 고정된 환경정보를 설정
2)SqlSessionFactoryBuilder
-MyBatis 설정 file을 바탕으로 SqlSessionFactory를 생성
3)SqlSessionFactory
-SqlSession을 생성
4)SqlSession
-핵심적인 역할을 하는 class로서 Sql 실행이나 transaction 관리를 실행
-SqlSession object는 Thread-Safe하지 않으므로 thread마다 필요에 따라 생성
5)Mapping File(user.xml)
-SQL문과 ORMapping을 설정

7. MyBatis-Spring의 주요 component의 역할
1)MyBatis 설정file(sqlMapConfig.xml)
-VO 객체의 정보를 설정
2)SqlSessionFactoryBean
-MyBatis 설정file을 바탕으로 SqlSessionFactory를 생성
-Spring Bean으로 등록해야 함.
3)SqlSessionTemplate
-핵심적인 역할을 하는 class로서 SQL 실행이나 transaction 관리를 실행한다.
-SqlSession interface를 구현하며, Thread-Safe하다.
-Spring Bean으로 등록해야 함.
4)Mapping File(mybatis-mapper.xml)
-SQL문과 OR Mapping을 설정
5)Spring Bean 설정file(beans.xml)
-SqlSessionFactoryBean을 Bean 등록할 때 DataSource 정보와 MyBatis Config file정보, Mapping file의 정보를 함께 설정한다.
-SqlSessionTemplate을 Bean으로 등록한다.

8. Lab : MySQL의 World Database의 City Table 가져오기
1)MariaDB 설치하기
-https://mariadb.org
-https://downloads.mariadb.org에서 MariaDB 10.3.15 Stable
-mariadb-10.3.15-winx64.msi

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

```
103
104   3)src > right-click > New > Package
105     -Package name : com.example
106
107   4)Java Project를 Spring Project로 변환
108     -MybatisDemo Project > right-click > Configuration > Convert to Maven Project
109     -Project : /MybatisDemo
110     -Group Id : MybatisDemo
111     -Artifact Id : MybatisDemo
112     -version : 0.0.1-SNAPSHOT
113     -Packaging : jar
114     -Finish
115
116     -MybatisDemo Project > right-click > Spring > Add Spring Project Nature
117
118     -pom.xml 파일에 Spring Context Dependency 추가하기
119       <version>0.0.1-SNAPSHOT</version>
120       <dependencies>
121         <dependency>
122           <groupId>org.springframework</groupId>
123           <artifactId>spring-context</artifactId>
124           <version>4.3.24.RELEASE</version>
125         </dependency>
126       </dependencies>
127
128     -pom.xml > right-click > Run As > Maven install
129       [INFO] BUILD SUCCESS 확인
130
131   5)MyBatis library 검색 및 설치
132     -Maven Repositoy에서 'mybatis'로 검색
133
134       <dependency>
135         <groupId>org.mybatis</groupId>
136         <artifactId>mybatis</artifactId>
137         <version>3.5.1</version>
138       </dependency>
139
140     -pom.xml에 등록 및 설치
141
142   6)MyBatis-Spring library 검색 및 설치
143     -Maven Repository에서 'mybatis spring'으로 검색
144
145       <dependency>
146         <groupId>org.mybatis</groupId>
147         <artifactId>mybatis-spring</artifactId>
148         <version>2.0.1</version>
149       </dependency>
150
151     -pom.xml에 등록 및 설치
152
153   7)MariaDB Jdbc Driver library 검색 및 설치
154     -Maven Repository 에서 'mariadb'로 검색하여 MariaDB Java Client를 설치한다.
155
156         <dependency>
```

```
157             <groupId>org.mariadb.jdbc</groupId>
158             <artifactId>mariadb-java-client</artifactId>
159             <version>2.4.1</version>
160         </dependency>
161
162     -pom.xml에 붙여 넣고 Maven Install 하기
163
164   8)Spring JDBC 설치
165     -JdbcTemplate를 사용하기 위해 pom.xml에 다음 dependency를 추가해야 함.
166
167         <dependency>
168           <groupId>org.springframework</groupId>
169           <artifactId>spring-jdbc</artifactId>
170           <version>4.3.24.RELEASE</version>
171         </dependency>
172
173     -pom.xml에 붙여 넣고 Maven Install 하기
174
175   9)jUnit Library 설치
176     -http://mvnrepository.com에 접근
177     -jUnit으로 검색
178     -jUnit 4.12 version을 pom.xml에 추가
179
180       <dependency>
181         <groupId>junit</groupId>
182         <artifactId>junit</artifactId>
183         <version>4.12</version>
184         <scope>test</scope>
185       </dependency>
186
187     -pom.xml > right-click > Run As > Maven Install
188
189   10)MybatisDemo/resources folder 생성
190     -MybatisDemo project > right-click > Build Path > Configure Build Path
191     -Source Tab > Add Folder
192     -MybatisDemo click
193     -Create New Folder > Folder name : resources > Finish > OK
194     -MybatisDemo/resources(new) 확인
195     -Apply and Close
196
197   11)resources/dbinfo.properties file 생성
198     -/resoures/dbinfo.properties file 생성
199     -/resources > right-click > New > File
200     -File name : dbinfo.properties > Finish
201
202       db.driverClass=org.mariadb.jdbc.Driver
203       db.url=jdbc:mariadb://localhost:3306/world
204       db.username=root
205       db.password=javamariadb
206
207   12)world database downloads
208     -https://dev.mysql.com/doc/index-other.html
209     -Example Databases > world database > Zip
210     -Unzip
```

```
211     -MariaDB login 후 world database 실행
212
213   13)여러 Package 생성
214     -/src/com.example.vo
215     -/src/com.example.service
216     -/src/com.example.dao
217
218   14)VO class 작성
219     -/src/com.example.vo.CityVO.java 생성
220
221       package com.example.vo;
222
223       public class CityVO {
224         private int id;
225         private String name;
226         private String countryCode;
227         private String district;
228         private int population;
229         public int getId() {
230           return id;
231         }
232         public void setId(int id) {
233           this.id = id;
234         }
235         public String getName() {
236           return name;
237         }
238         public void setName(String name) {
239           this.name = name;
240         }
241         public String getCountryCode() {
242           return countryCode;
243         }
244         public void setCountryCode(String countryCode) {
245           this.countryCode = countryCode;
246         }
247         public String getDistrict() {
248           return district;
249         }
250         public void setDistrict(String district) {
251           this.district = district;
252         }
253         public int getPopulation() {
254           return population;
255         }
256         public void setPopulation(int population) {
257           this.population = population;
258         }
259         @Override
260         public String toString() {
261           return String.format("CityInfoVO [id=%s, name=%s, countryCode=%s, district=%s,
              population=%s]", id, name,
262                 countryCode, district, population);
263         }
```

```
264       }
265
266   15)Mapping file 작성 및 MyBatis 설정
267     -/resources/SqlMapConfig.xml
268
269         <?xml version="1.0" encoding="UTF-8" ?>
270         <!DOCTYPE configuration
271           PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
272             "http://mybatis.org/dtd/mybatis-3-config.dtd">
273         <configuration>
274           <typeAliases>
275             <typeAlias type="com.example.vo.CityVO" alias="cityVO" />
276           </typeAliases>
277
278         </configuration>
279
280     -/resources/mybatis-mapper.xml
281
282         <?xml version="1.0" encoding="UTF-8"?>
283         <!DOCTYPE mapper
284           PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
285             "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
286         <mapper namespace="City">
287         <resultMap id="cityResult" type="cityVO">
288           <result property="id" column="ID" />
289           <result property="name" column="Name" />
290           <result property="district" column="District" />
291           <result property="countryCode" column="CountryCode" />
292           <result property="population" column="Population" />
293         </resultMap>
294
295         <select id="selectCityByName" parameterType="String" resultType="cityVO"
          resultMap="cityResult">
296           SELECT * FROM world.city WHERE  name = #{name}
297         </select>
298       </mapper>
299
300   16)Dao 객체 생성
301     -/src/com.example.dao.CityDao.java
302
303       package com.example.dao;
304
305       import java.util.List;
306       import com.example.vo.CityVO;
307
308       public interface CityDao {
309         List<CityVO> readAll();
310         CityVO read(String name);
311       }
312
313     -/src/com.example.dao.CityDaoImpl.java
314
315       package com.example.dao;
316
```

```
317         import java.util.List;
318
319         import org.apache.ibatis.session.SqlSession;
320         import org.springframework.beans.factory.annotation.Autowired;
321         import org.springframework.stereotype.Repository;
322
323         import com.example.vo.CityVO;
324         @Repository("cityDao")
325         public class CityDaoImpl implements CityDao {
326
327           @Autowired
328             private SqlSession session;
329
330           @Override
331           public List<CityVO> readAll() {
332             return null;
333           }
334
335           @Override
336           public CityVO read(String name) {
337             CityVO city  = session.selectOne("City.selectCityByName", name);
338             return city;
339           }
340         }
341
342     17)Service 객체 생성
343       -/src/com.example.service.CityService.java
344
345           package com.example.service;
346
347           import java.util.List;
348           import com.example.vo.CityVO;
349
350           public interface CityService {
351             List<CityVO> getCityList();
352             CityVO getCity(String name);
353           }
354
355       -/src/com.example.service.CityServiceImpl.java
356           package com.example.service;
357
358           import java.util.List;
359
360           import org.springframework.beans.factory.annotation.Autowired;
361           import org.springframework.stereotype.Service;
362
363           import com.example.dao.CityDao;
364           import com.example.vo.CityVO;
365
366           @Service("cityService")
367           public class CityServiceImpl implements CityService {
368
369             @Autowired
370             CityDao cityDao;
```

```
371
372           @Override
373           public List<CityVO> getCityList() {
374             return null;
375           }
376
377           @Override
378           public CityVO getCity(String name) {
379             return this.cityDao.read(name);
380           }
381         }
382
383   18)Bean Configuration XML 작성
384     -/resources > right-click > New > Spring Bean Configuration File
385     -File name : beans.xml > Finish
386     -Namespace Tab
387     -Check context - http://www.springframework.org/schema/context
388
389       <?xml version="1.0" encoding="UTF-8"?>
390       <beans xmlns="http://www.springframework.org/schema/beans"
391         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
392         xmlns:context="http://www.springframework.org/schema/context"
393         xsi:schemaLocation="http://www.springframework.org/schema/beans
          http://www.springframework.org/schema/beans/spring-beans.xsd
394           http://www.springframework.org/schema/context
              http://www.springframework.org/schema/context/spring-context-3.2.xsd">
395
396       <!-- mybatis-spring 설정 -->
397       <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
398         <property name="dataSource"  ref="dataSource" />
399         <property name="configLocation"   value="classpath:SqlMapConfig.xml" />
400         <property name="mapperLocations">
401           <list>
402             <value>classpath:mybatis-mapper.xml</value>
403           </list>
404         </property>
405       </bean>
406
407       <bean id="sqlSession" class="org.mybatis.spring.SqlSessionTemplate">
408         <constructor-arg ref="sqlSessionFactory" />
409       </bean>
410
411       <context:property-placeholder location="classpath:dbinfo.properties" />
412       <bean id="dataSource"
          class="org.springframework.jdbc.datasource.SimpleDriverDataSource">
413         <property name="driverClass" value="${db.driverClass}" />
414         <property name="url" value="${db.url}" />
415         <property name="username" value="${db.username}" />
416         <property name="password" value="${db.password}" />
417       </bean>
418     </beans>
419
420   19)사용자 관리 project의 Bean 등록 및 의존 관계 설정
421     -<context:component-scan> tag 사용
```

-@Service, @Repository annotation을 선언한 class들과 @Autowired annotation을 선언하여 의존관계를 설정한 class들이 위치한 package를 Scan하기 위한 설정을 XML에 해주어야 한다.
-beans.xml에 다음 code 추가한다.

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

20)Spring TestContext Framework 사용하기
-/src/com.example.test package 생성
-/src/com.example.test > right-click > New > JUnit Test Case
-Name : MybatisDemoTest > Finish
-New JUnit Test Case창에서 [Not now] 선택 > OK

```
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);
  }
}
```

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

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

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

-com.example.dao.CityDaoImpl.java

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

```
474       }
475
476     -mybatis-mapper.xml
477
478       <select id="selectList" resultType="cityVO" resultMap="cityResult">
479         SELECT * FROM world.city ORDER BY id DESC
480       </select>
481
482     -MybatisDemoTest.java
483
484       @Autowired
485       CityService cityService;
486
487       @Ignore @Test
488       public void test() {
489         CityVO city = this.cityService.getCity("Seoul");
490         System.out.println(city);
491       }
492
493       @Test
494       public void test1() {
495         List<CityVO> list = this.cityService.getCityList();
496
497         for(CityVO vo : list){
498           System.out.println(vo.getId());
499           System.out.println(vo.getName());
500           System.out.println(vo.getDistrict());
501           System.out.println(vo.getCountryCode());
502           System.out.println(vo.getPopulation());
503           System.out.println("-----------------------");
504         }
505       }
506
507   23)Test
508     -right-click > Run As > Junit Test
509     -결과 -> Junit View에 초록색 bar
```