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

2. 특징
1)쉬운 접근성과 코드의 간결함
-가장 간단한 persistence 프레임워크이다.
-XML 형태로 서술된 JDBC 코드라고 생각해도 될 만큼 JDBC의 모든 기능을 MyBatis가 대부분 제공한다.
-복잡한 JDBC 코드를 걷어내며 깔끔한 소스코드를 유지할 수 있다.
-수동적인 파라미터 설정과 쿼리 결과에 대한 맵핑 구문을 제거할 수 있다.

2)SQL 문과 프로그래밍 코드의 분리
-SQL에 변경이 있을 때마다 자바 코드를 수정하거나 컴파일 하지 않아도 된다.
-SQL 작성과 관리 또는 검토를 DBA와 같은 개발자가 아닌 다른 사람에게 맡길 수도 있다.

3)다양한 프로그래밍 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 컴포넌트
-그림참조

-http://terasolunaorg.github.io/guideline/5.0.0.RELEASE/en/ArchitectureInDetail/DataAccessMyBatis3.html

6. MyBatis 3의 주요 컴포넌트의 역할
   1)MyBatis 설정파일(SqlMapconfig.xml)
      -데이터베이스의 접속 주소 정보나 Mapping 파일의 경로 등의 고정된 환경정보를 설정
   2)SqlSessionFactoryBuilder
      -MyBatis 설정 파일을 바탕으로 SqlSessionFactory를 생성
   3)SqlSessionFactory
      -SqlSession을 생성
   4)SqlSession
      -핵심적인 역할을 하는 클래스로서 Sql 실행이나 트랜잭션 관리를 실행
      -SqlSession 오브젝트는 Thread-Safe하지 않으므로 thread마다 필요에 따라 생성
   5)Mapping File(user.xml)
      -SQL문과 ORMapping을 설정

7. MyBatis-Spring의 주요 컴포넌트의 역할
   1)MyBatis 설정파일(sqlMapConfig.xml)
      -VO 객체의 정보를 설정
   2)SqlSessionFactoryBean
      -MyBatis 설정파일을 바탕으로 SqlSessionFactory를 생성
      -Spring Bean으로 등록해야 함.
   3)SqlSessionTemplate
      -핵심적인 역할을 하는 클래스로서 SQL 실행이나 트랜잭션 관리를 실행한다.
      -SqlSession 인터페이스를 구현하며, Thread-Safe하다.
      -Spring Bean으로 등록해야 함.
   4)Mapping File(mybatis-mapper.xml)
      -SQL문과 OR Mapping을 설정
   5)Spring Bean 설정파일(beans.xml)
      -SqlSessionFactoryBean을 Bean 등록할 때 DataSource 정보와 MyBatis Config 파일정보, Mapping 파일의 정보를 함께 설정한다.
      -SqlSessionTemplate을 Bean으로 등록한다.

8. Lab : MySQL의 World Database의 City Table 가져오기
   1)MyBatisDemo project 생성
      -Spring Legacy Project
      -Simple Maven Project

   2)MyBatis library 검색 및 설치
      -Maven Repositoy에서 'mybatis'로 검색

```
<dependency>
      <groupId>org.mybatis</groupId>
      <artifactId>mybatis</artifactId>
      <version>3.4.5</version>
</dependency>
```

      -pom.xml에 등록 및 설치

   3)MyBatis-Spring library 검색 및 설치
      -Maven Repository에서 'mybatis spring'으로 검색

```
<dependency>
      <groupId>org.mybatis</groupId>
      <artifactId>mybatis-spring</artifactId>
      <version>1.3.1</version>
</dependency>
```

      -pom.xml에 등록 및 설치

   4)MySQL Jdbc Driver 라이브러리 검색 및 설치
      -Maven Repository 에서 'spring mysql'로 검색하여 MySQL Connector/J를 설치한다.

```
<dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>5.1.42</version>
```

```
        </dependency>

   -pom.xml에 붙여 넣고 Maven Install 하기

5)Spring JDBC 설치
   -JdbcTemplate를 사용하기 위해 pom.xml에 다음 dependency를 추가해야 함.

        <dependency>
           <groupId>org.springframework</groupId>
           <artifactId>spring-jdbc</artifactId>
           <version>4.3.9.RELEASE</version>
        </dependency>

   -pom.xml에 붙여 넣고 Maven Install 하기

6)dbinfo.properties 파일 생성
   -/src/main/resoures/dbinfo.properties 파일 생성
   -/src/main/resources > right-click > New > File
   -File name : dbinfo.properties > Finish

      db.driverClass=com.mysql.jdbc.Driver
      db.url=jdbc:mysql://192.168.56.15:3306/world
      db.username=root
      db.password=javamysql

7)여러 Package 생성
   -/src/main/java/com.javasoft.vo
   -/src/main/java/com.javasoft.service
   -/src/main/java/com.javasoft.dao

8)여러 클래스 작성
   -/src/main/java/com.javasoft.CityVO.java 생성

        package com.javasoft;

        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() {
              return name;
           }
           public void setName(String name) {
              this.name = name;
           }
           public String getCountryCode() {
              return countryCode;
           }
           public void setCountryCode(String countryCode) {
              this.countryCode = countryCode;
           }
           public String getDistrict() {
              return district;
           }
           public void setDistrict(String district) {
              this.district = district;
           }
           public int getPopulation() {
              return population;
```

```
195             }
196             public void setPopulation(int population) {
197                this.population = population;
198             }
199             @Override
200             public String toString() {
201                return String.format("CityInfoVO [id=%s, name=%s, countryCode=%s, district=%s,
                   population=%s]", id, name,
202                      countryCode, district, population);
203             }
204          }
205
206       -/com.javasoft.dao.CityDao.java
207
208          package com.javasoft.dao;
209
210          import java.util.List;
211          import com.javasoft.vo.CityVO;
212
213          public interface CityDao {
214             List<CityVO> readAll();
215             CityVO read(String name);
216          }
217
218       -/com.javasoft.dao.CityDaoImpl.java
219
220          package com.javasoft.dao;
221
222          import java.util.List;
223
224          import org.apache.ibatis.session.SqlSession;
225          import org.springframework.beans.factory.annotation.Autowired;
226          import org.springframework.stereotype.Repository;
227
228          import com.javasoft.vo.CityVO;
229          @Repository("cityDao")
230          public class CityDaoImpl implements CityDao {
231
232             @Autowired
233                private SqlSession session;
234
235             @Override
236             public List<CityVO> readAll() {
237                return null;
238             }
239
240             @Override
241             public CityVO read(String name) {
242                CityVO city  = session.selectOne("City.selectCityByName", name);
243                return city;
244             }
245          }
246
247       -/com.javasoft.service.CityService.java
248
249          package com.javasoft.service;
250
251          import java.util.List;
252          import com.javasoft.vo.CityVO;
253
254          public interface CityService {
255             public List<CityVO> getCityList();
256             public CityVO getCity(String name);
257          }
258
259       -/com.javasoft.service.CityServiceImpl.java
260          package com.javasoft.service;
```

```
import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.javasoft.dao.CityDao;
import com.javasoft.vo.CityVO;

@Service("cityService")
public class CityServiceImpl implements CityService {

   @Autowired
   CityDao citydao;

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

   @Override
   public CityVO getCity(String name) {
      return citydao.read(name);
   }
}
```

9)Mapping 파일 작성 및 MyBatis 설정

-/src/main/resources/SqlMapConfig.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>
   <typeAliases>
      <typeAlias type="com.javasoft.vo.CityVO" alias="cityVO" />
   </typeAliases>

</configuration>
```

-/src/main/resources/mybatis-mapper.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="City">
<resultMap id="cityResult" type="cityVO">
   <result property="id" column="ID" />
   <result property="name" column="Name" />
   <result property="district" column="District" />
   <result property="countryCode" column="CountryCode" />
   <result property="population" column="Population" />
</resultMap>

<select id="selectCityByName" parameterType="String" resultType="cityVO"
resultMap="cityResult">
   SELECT * FROM world.city WHERE  name = #{name}
</select>
  </mapper>
```

10)Bean Configuration XML 작성

-/src/main/resources > right-click > New > Spring Bean Configuration File

-File name : beans.xml > Finish

-Namespace Tab

-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>
```

11)사용자 관리 프로젝트의 Bean 등록 및 의존 관계 설정
-<context:component-scan> 태그 사용
-@Service, @Repository 어노테이션을 선언한 클래스들과 @Autowired 어노테이션을 선언하여 의존관계를 설정한
클래스들이 위치한 패키지를 Scan하기 위한 설정을 XML에 해주어야 한다.
-beans.xml에 다음 코드 추가한다.

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

12)Spring TestContext Framework 사용하기
-/src/test/java > right-click > New > JUnit Test Case
-Name : MyBatisDemoTest > 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.javasoft.service.CityService;
import com.javasoft.vo.CityVO;

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

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

```
        }

13)All City 읽어오기
  -com.javasoft.service.CityServiceImpl.java

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

  -com.javasoft.dao.CityDaoImpl.java

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

  -mybatis-mapper.xml

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

  -MyBatisDemoTest.java

    @Autowired
    CityService service;

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

    @Test
    public void test1() {
      List<CityVO> list = service.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("-----------------------");
      }
    }
```