http://www.egovframe.go.kr/wiki/doku.php?id=egovframework:rte2:psl:dataaccess:ibatis_vs_mybatis

Notable Changes in MyBatis

The notable changes in iBATIS from the conventional MyBatis are as follows:

Change in Terms (General)

iBatis	MyBatis	Remark
com.ibatis.*	org.apache.ibatis.*	Change in Package Structures
SqlMapConfig	Configration	Change in Terms
sqlMap	mapper	Change in Terms
sqlMapClient	sqlSession	Syntax Replaced
rowHandler	resultHandler	Syntax Replaced
resultHandler	SqlSessionFactory	Syntax Replaced
parameterMap, parameterClass	parameterType	Property Integrated
resultClass	resultType	Change in Terms
#var#	#{var}	Syntax Replaced
var	${var}	Syntax Replaced
<isEqual> , <isNull>	<if>	Syntax Replaced

Changes

Change in Package Structures

iBatis	MyBatis
com.ibatis.*	org.apache.ibatis.*

Those structurally changed iBATIS packages still keep their original package names.

MyBatis Library Provided

Maven Dependency Information Examples

```
<dependency>
	<groupId>org.mybatis</groupId>
	<artifactId>mybatis</artifactId>
	<version>3.2.2</version>
</dependency>

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

<version>1.2.0</version>
</dependency>

Annotation Introduced

- With the annotation available, sqlMapClient DI configuration for DAO is no longer required.
- You can use annotation in Spring 2.5 or newer.
- With annotation, all you need to do is to designate the specific bean id sqlSessionFactory and sqlSessionTemplate.

rowHandler no longer required

- Even with processing of xml and massive data, rowHandler is no longer required.
- sqlMapClient is now replaced by sqlSession, where API provides the specific method intended for large data processing.
- Also notable is that rowHandler is now replaced by resultHandler.
- This means that Java Annotation for xml is no longer required for what only takes Java now. You cal also declare DataSource, Environment, etc. in Configration.xml for classification.
- Keep in mind, however, that if you have established configuration via xml and classified environment variables and properties, these variables and properties are to be overwritten by the Java Class declaration. It is thus advised that you do not work on a multiple methods when you want to create a project.
- When you declare "configuration configuration = new Con….", you'll need to add configuration.addMapper(UserMapper.class) as mapper in no longer xml-based. You thus need to make sure you choose either one of them before creating a project.

Type of Namespace Changed

- Name of sqlMap is now be used in full (in the user manual you can find the name of sqlMap is 'Good' as the author intends to make things clear).

iBatis **MyBatis**

<sqlMap namespace="User"> <mapper namespace="myBatis.mapper.UserMapper">

You'll need to make an extensive call when intending to call from the Java side.

list = session.selectList("myBatis.mappers.UserMapper.getUserList");

It is advised that you create the mapper file in the form of Java, not the conventional xml format using the Java annotation (@Select).

UserMapper mapper = session.getMapper(UserMapper.class);

list = mapper.selectUserList();

Properties Changed or Added

You would see some new tags that are intended to be used to create queries changing by conditions given. Tags are now much more intuitive and fit the situation given (Update, Select).

- You would no longer use parameterMap. parameterMap and parameterClass are now replaced by parameterType.
- With resultMap still available, resultClass is also replaced by resultType.
- You can use parameterType and resultType as they used to be, from basic types (int, byte, ….) to class names.
- Property "statementType" now replaces <procedure>, where options PREPARED, STATEMENT and CALLABLE are available. The default option is PREPARED.
- Intended for mapping the parameters, #var# is now replaced by #{var}. Note var is also replaced by ${var}.

Remark) #{var} and ${var}are used as parameters of prepredStatement and string, respectively. You may add "order by" following $, like ${orderParam}, in which case the pre-determined value by the developer must be input for MyBatis to help determine adequacy of quaries.

typeAlias in sqlMap is no longer used in the mapper with altered sqlMap and now defined in the Configration file.

```
<typeAliases>
      <typeAlias type="vo.UserVO" alias="User"/>
</typeAliases>
```

You can define Alias in the Configration file for use in the entire mapper.

Dynamic Statement Changed

- Syntaxes <isEqual>, <isNull> and equivalent are now replaced by <if>.

The syntax has thus been simplified, like <if test="userID != null">. With the change, you can also use <set> in <where> or update. This replaces the conventional <dynamic> statements of WHERE, AND and OR.

```
<select id="getUserList" resultType="User>
      SELECT * FROM TR_USER
            <where>
                  <if test="isAdmin != null">
                        authLevel = '1'
                  </if>
            </where>
</select>
```

- Tags trim and foreach are added.

1) Tag "trim" is used when you want to trim off the disqualified strings after comma (,) for dynamic creation of a string of queries.
2) Tag "foreach" is used when the repetitive items are dynamically input. (e.g. use of "in" in WHERE statement)

References

- MyBatis Migration Guideline (http://code.google.com/p/mybatis/wiki/DocUpgrade3)
- http://blog.naver.com/PostView.nhn?blogId=vikong&logNo=60180433051