

myBatis

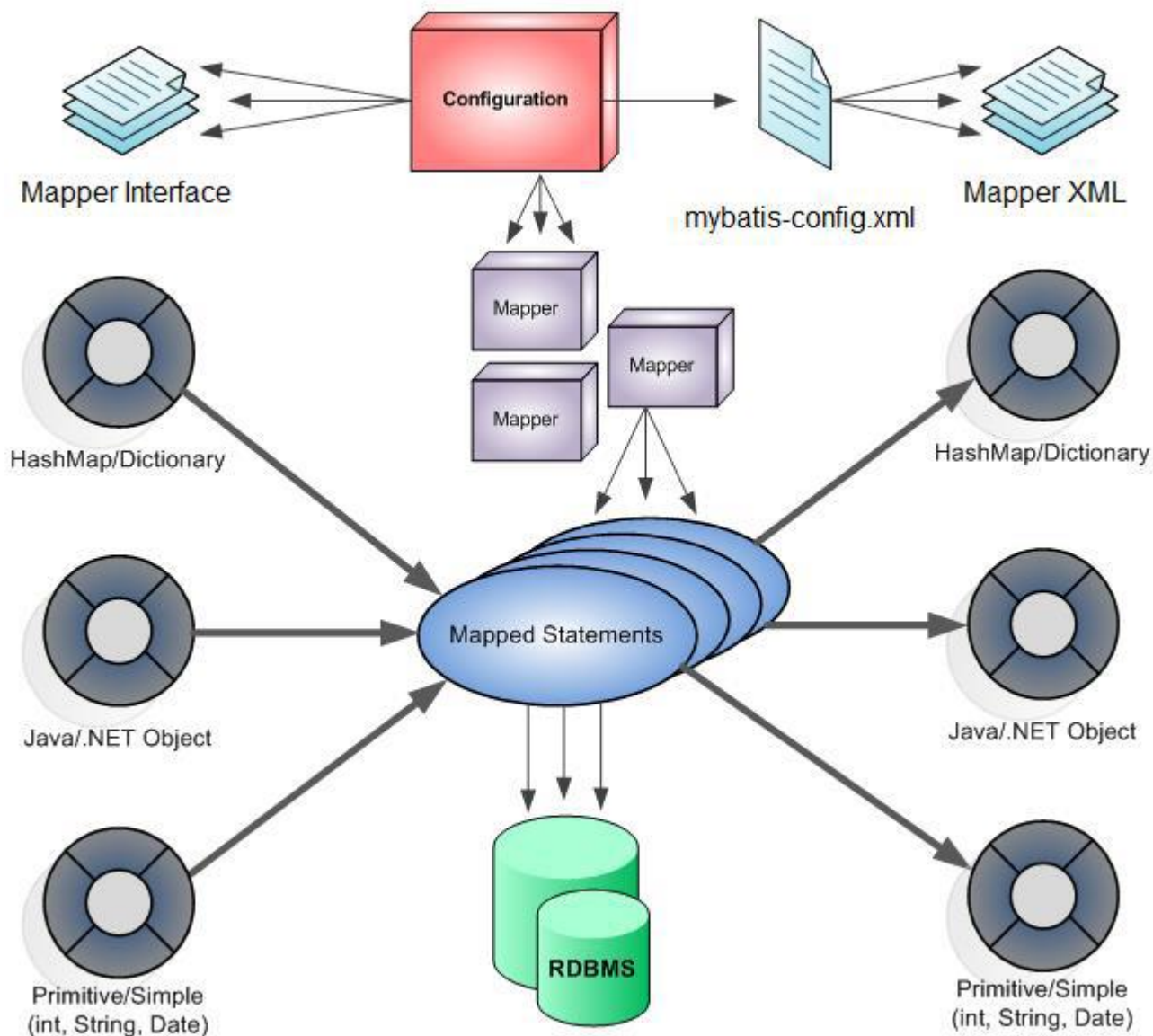
myBatis

- ✓ myBatis의 정의 및 특성
 - ✓ myBatis API 및 SQL 관련 XML 태그
 - ✓ 실습
-

myBatis 정의 및 특성

- ✓ 2010년에 iBatis가 Apache를 떠나 Google Code 로 이동하면서 이름이 myBatis로 변경됨.
 - ✓ 더 빠른 JDBC 코딩을 위한 일반화된 프레임워크.
 - ✓ SQL을 자바 코드가 아닌 XML로 따로 분리
 - ✓ SQL의 실행 결과를 Map 또는 자바 클래스로 자동 매핑
 - ✓ SQL을 XML 이나 인터페이스내에 어노테이션을 활용하여 처리
-

myBatis 정의 및 특성



myBatis 적용

- ✓ 환경설정 파일(**SqlMapConfig**)과 실제 쿼리를 적용할 파일 (**Mapper**)이 필요(**XML, Annotation**)
 - ✓ 자바파일에서는 설정된 환경설정 정보를 이용하여 **SqlSession**객체를 얻어온 다음 **myBatis**와 연동
-

myBatis 다운로드

- ✓ <https://github.com/mybatis/mybatis-3/releases> 다운로드
 - ✓ 프로젝트에 다운받은 파일을 압축 해제 한 후 **mybatis-x.x.x.jar** 파일을 클래스패스에 추가(웹 프로젝트인 경우 **WEB-INF/lib**에 추가)
 - ✓ **Maven** 일 경우 **Dependency** 추가
-

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>x.x.x</version>

</dependency>

xml 환경 설정

✓ 설정 루트

```
<configuration>
```

- vo 객체의 타입 Alias 설정

```
<typeAliases>
```

```
    <typeAlias alias="member" type="member.Member" />
```

```
</typeAliases>
```

- 설정파일 내부에서 사용하는 값을 파일을 활용하여 추출

```
<properties resource="db.properties" />
```

```
..... 생략...
```

```
</configuration>
```

xml 환경 설정

`<configuration>`

..... 생략...

- 트랜잭션 및 데이터베이스 설정

`<environments default="development">`

`<environment id="development">`

`<transactionManager type="JDBC"/>`

트랜잭션

`<dataSource type="POOLED">` UNPOOLED, POOLED, JNDI

`<property name="driver" value="..."/>`

`<property name="url" value="..."/>`

`<property name="username" value="hr"/>`

`<property name="password" value="hr"/>`

`</dataSource>`

데이터베이스

`</environment>`

`</environments>`

`</configuration>`

xml 환경 설정

- 매퍼정보 설정

```
<mappers>
```

```
    <mapper resource="common/db/member.xml"/>
```

```
</mappers>
```

```
</configuration>
```

XML Mapper 사용 태그

구문 형식	속성	하위 요소	사용 용도
<select>	id parameterType resultType resultMap	모든 동적 요소	데이터 조회
<sql>	id		공통 쿼리 묶기
<include>	refid		<sql> 로 생성된 컴포넌트 삽입
<insert>	id parameterType	모든 동적 요소 <selectKey>	데이터 입력
<update>	id parameterType	모든 동적 요소	데이터 수정
<delete>	id parameterType	모든 동적 요소	데이터 삭제

XML Mapper 동적 태그

구문형식	속성	사용법
<foreach>	Collection Item Open Separator close	<pre><foreach item="addr" open="(" separator="," close=")" collection="addrs"> #{addr} </foreach></pre>
<if>	test	<pre><if test="id != null">id = #{id}</if></pre>
<choose> <when> <otherwise>	test	<pre><choose> <when test="searchType == 'id'"> id like #{searchWord} </when> <otherwise> addr like #{searchWord} </otherwise> </choose></pre>

XML Mapper 동적 태그

구문형식	속성	사용법
<where>		<pre><where> <if test="writer != null"> writer like #{writer} </if> <if test="title != null"> AND title like #{title} </if> </where></pre>
<set>		<pre><set> <if test="writer != null"> writer = #{writer}, </if> <if test="title != null"> title = #{title} </if> </set></pre>

SqlSession 클래스 중요 메소드

이름	설명
java.util.List	selectList(String id)
java.util.List	selectList(String id, Object parameterObject)
java.lang.Object	selectOne(String id)
java.lang.Object	selectOne(String id, Object parameterObject)
int	insert(String id, Object parameter)
int	delete(String id, Object parameter)
int	update(String id, Object parameter)
