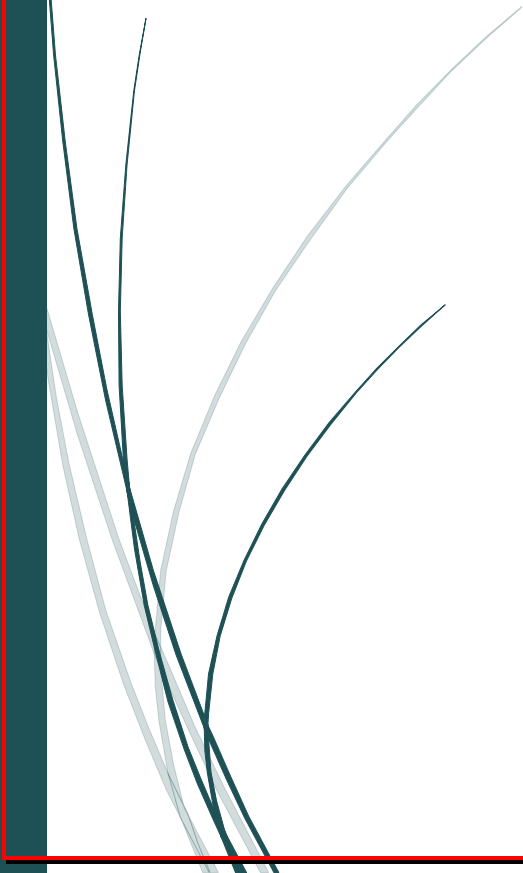


The Spring logo, which is a red arrow pointing to the right with the word "Spring" written inside in white.

Spring

MyBatis

Several thin, curved lines in shades of teal and grey extending from the left edge of the page towards the center.

권기웅



1. 설치

1.1 .jar 다운로드

<http://blog.mybatis.org/>

The screenshot shows the MyBatis blog page at blog.mybatis.org/p/products.html. The page features the MyBatis logo and a navigation menu with 'Products' highlighted. The 'Products' section contains a table listing various projects and their download links. The 'download' link for 'MyBatis 3' is highlighted with a red box.

Project	Description	Links
MyBatis 3	SQL Mapping Framework for Java	download docs
Generator	Code generator for MyBatis and iBATIS	download docs eclipse plugin
Migrations	Migrations database change management tool	download docs
Migrations Maven Plugin	Maven plugin for Migrations	download docs
MyBatispe	Eclipse plugin for MyBatis	download docs
MyBatis for Scala	MyBatis for Scala	download docs

Integration

Project	Description	Links
Spring	Integration with Spring	download docs
Guice	Integration with Guice	download docs
CDI	Integration with CDI	download docs

MyBatis for Java

- Github Project
- Java Mailing List
- China Mailing List

MyBatis.NET

- .NET Google Code Project
- .NET Mailing List

Blog Archive

- 2015 (2)
 - May (1)
 - MyBatis 3.3.0 released!
 - March (1)
- 2014 (14)
- 2013 (10)
- 2012 (11)





Releases · mybatis/mybatis-3 · GitHub, Inc. [US] | <https://github.com/mybatis/mybatis-3/releases>

Latest release

mybatis-3.3.0
22fcbef

mybatis-3.3.0

emacarron released this 24 days ago · 7 commits to master since this release

Includes two main changes:

- Ognl has been upgraded to 3.0.11 that is the latest version
- The default proxy tool is now Javassist and is included inside the mybatis jar

There are some other minor changes and bug fixes. See the details at <https://github.com/mybatis/mybatis-3/issues?q=milestone%3A3.3.0>

Note that you can still use CGLIB as the proxy factory tool by adding CGLIB to your classpath and setting the MyBatis global parameter as follows:

```
<settings>
  <setting name="proxyFactory" value="CGLIB"/>
</settings>
```

Downloads

mybatis-3.3.0.zip	3.04 MB
Source code (zip)	
Source code (tar.gz)	

1.2 문서

<http://mybatis.github.io/mybatis-3/ko/index.html>

MyBatis - MyBatis 3 | 소개 x

mybatis.github.io/mybatis-3/ko/index.html

mybatis

Last Published: 24 5월 2015 | Version: 3.3.0

CORE

- 소개
- 시작하기
- 매퍼 설정
- Mapper XML 파일
- 동적 SQL
- 자바 API
- 구문 빌더
- 로깅
- 프로젝트 문서화
- 프로젝트 정보
- 프로젝트 보고서

소개

MyBatis 는 무엇인가?

MyBatis 는 개발자가 지정한 SQL, 저장프로시저 그리고 몇가지 고급 매핑을 지원하는 퍼시스턴스 프레임워크이다. MyBatis 는 JDBC 코드와 수동으로 셋팅하는 파라미터와 결과 매핑을 제거한다. MyBatis 는 데이터베이스 레코드에 원시타입과 Map 인터페이스 그리고 자바 POJO 를 설정하고 매핑하기 위해 XML 과 애노테이션을 사용할 수 있다.

이 문서가 좀더 좋아지도록 도와주세요...

이 문서에서 빈약하거나 기능 중 일부의 내용이 없는 경우, MyBatis 에 대해 배우는 가장 좋은 방법은 당신 스스로 문서를 작성하는 것이다!

이 문서는 xdoc 포맷으로 되어 있으며, 프로젝트 GitHub 에서 볼 수 있다. 저장소를 포크합니다, 를 업데이트합니다, 그리고 풀 요청을 보냅니다.

당신은 이 문서의 가장 멋진 저자가 되는 셈이다. 많은 사람들이 이 문서를 읽을 것이다!

번역에 대하여

번역자 : 이동국(fromm0@gmail.com, <http://ldg.pe.kr>, <https://www.facebook.com/dongguk.lee.3>)

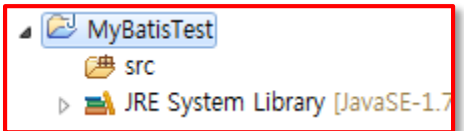
Built by:





2. 자바 프로젝트

2.1 자바 프로젝트 작성

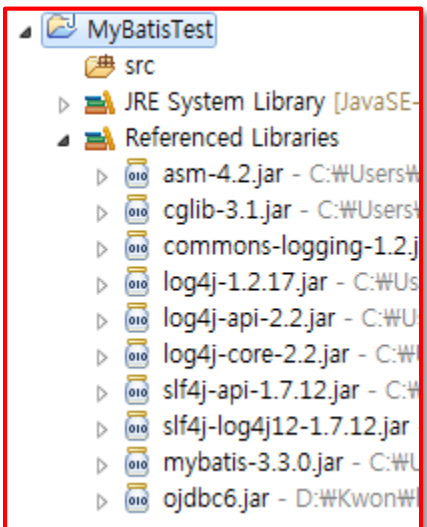


2.2 .jar 추가

ojdbc6.jar

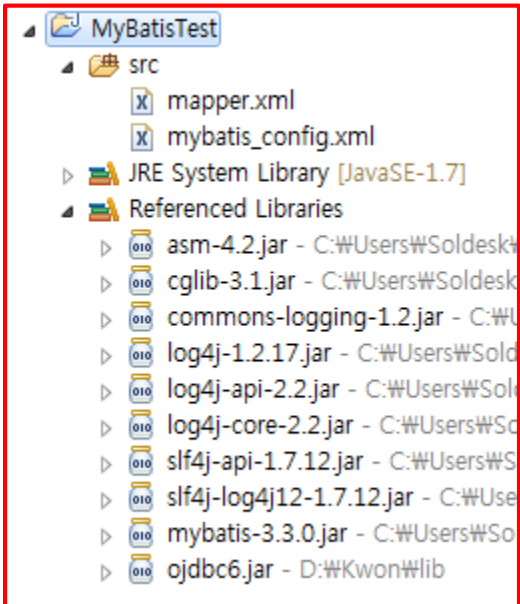
마이바티스홈/*.jar

마이바티스홈/lib/*.jar 추가





3.XML 파일





3.1 설정 XML 파일(config) - 연결정보/SQL XML 파일명

3.1.1 config 파일 DOCTYPE

```
<!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-config.dtd">
```

3.2.2 config 파일 내용

mybatis_config.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>
  <environments default="development">
    <environment id="development">
      <transactionManager type="JDBC" />
      <dataSource type="POOLED">
        <property name="driver" value="oracle.jdbc.driver.OracleDriver" />
        <property name="url" value="jdbc:oracle:thin:@172.16.101.141:1521:xe" />
        <property name="username" value="kwon" />
        <property name="password" value="kwon" />
      </dataSource>
    </environment>
  </environments>
  <mappers>
    <mapper resource="mapper.xml" />
  </mappers>
</configuration>
```





3.2 SQL XML(mapper) 파일

3.2.1 mapper 파일 DOCTYPE

```
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
```

3.2.2 mapper 파일 내용

```
<?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="kwon">
    <select id="selectProduct" resultType="com.kwon.Product">
        select * from jun11_product_kwon
    </select>
</mapper>
```

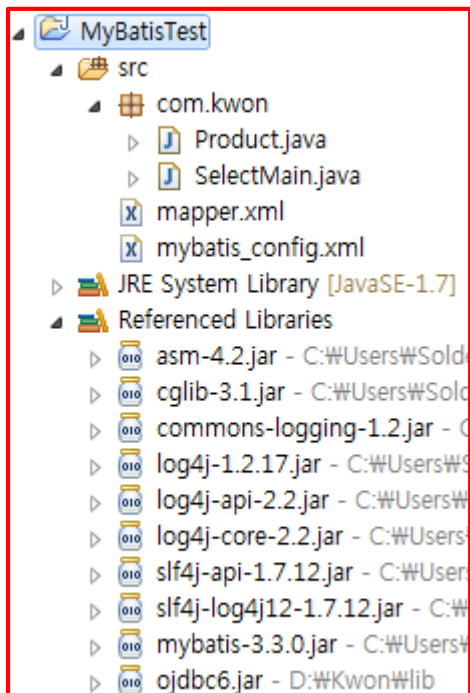




4. 자바 프로그램

SQL> desc jun11_product_kwon

Name	Null?	Type
P_ID	NOT NULL	CHAR(3 CHAR)
P_NAME		VARCHAR2(10 CHAR)
P_PRICE		NUMBER(7)





4.1 SELECT

4.1.1 자바빈

```
package com.kwon;

import java.math.BigDecimal;

public class Product {
    // 테이블 필드명과 변수명을 같게
    // 정수형은 BigDecimal타입으로
    private String p_id;
    private String p_name;
    private BigDecimal p_price;

    public Product(String p_id, String p_name, BigDecimal p_price) {
        super();
        this.p_id = p_id;
        this.p_name = p_name;
        this.p_price = p_price;
    }

    public String getP_id() {
        return p_id;
    }

    public void setP_id(String p_id) {
        this.p_id = p_id;
    }
}
```





4.1.2 SelectMain.java

```
public class SelectMain {
    public static void main(String[] args) {
        try {
            String cfgFileName = "mybatis_config.xml";
            InputStream cfgIs = Resources.getResourceAsStream(cfgFileName);

            SqlSessionFactory ssf = new SqlSessionFactoryBuilder().build(cfgIs);
            SqlSession ss = ssf.openSession();

            // [mapper namespace].[select id]
            List<Product> products = ss.selectList("kwon.selectProduct");
            for (Product p : products) {
                System.out.println(p.getP_id() + " " + p.getP_name() + " " + p.getP_price());
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

4.1.3 실행결과

```
mmk 메르스마스크 4000
tv1 옛날 100000
tv2 신형TV 500000
trs 한휴지통 2000
hcb HDMI케이블 5400
p01 컴퓨터 800000
p02 마우스 12000
p03 보드마카 1000
p04 쇼핑백 2000
p05 usb케이블 2000
p06 우산 5000
p08 의자 100000
p10 모니터 200000
mt 물통 300
```





4.2 INSERT

4.2.1 mapper.xml 에 추가

```
<?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="kwon">
    <select id="selectProduct" resultType="com.kwon.Product">
        select * from jun11_product_kwon
    </select>
    <insert id="insertProduct" parameterType="com.kwon.Product">
        insert into jun11_product_kwon
        values(#{p_id}, #{p_name}, #{p_price})
    </insert>
</mapper>
```





4.2.2 InsertMain.java

```
public class InsertMain {  
    public static void main(String[] args) {  
        try {  
            String cfgFileName = "mybatis_config.xml";  
            InputStream cfgIs = Resources.getResourceAsStream(cfgFileName);  
            SqlSessionFactory ssf = new SqlSessionFactoryBuilder().build(cfgIs);  
            SqlSession ss = ssf.openSession();  
  
            Scanner sc = new Scanner(System.in);  
            System.out.print("ID : ");  
            String p_id = sc.next();  
            System.out.print("품명 : ");  
            String p_name = sc.next();  
            System.out.print("가격 : ");  
            BigDecimal p_price = sc.nextBigDecimal();  
  
            Product p = new Product(p_id, p_name, p_price);  
  
            int row = ss.insert("kwon.insertProduct", p);  
  
            if(row == 1){  
                System.out.println("insert 성공");  
            }  
            ss.commit();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

4.2.3 실행결과

ID : mer
품명 : 메르스소독제
가격 : 1000000
insert 성공

p01 컴퓨터 800000
p02 마우스 12000
p03 보드마카 1000
p04 쇼핑백 2000
p05 usb케이블 2000
p06 우산 5000
p08 의자 100000
p10 모니터 200000
mt 물통 300
gkb 키보드 10000
mer 메르스소독제 1000000
rb1 르블랑 500
fbk 노란꽃가방 100000





4.3 UPDATE

4.3.1 mapper.xml 에 추가

```
<update id="updateProduct" parameterType="com.kwon.Product">
    update jun11_product_kwon
    set p_price=#{p_price}
    where p_id=#{p_id}
</update>
```

4.3.2 UpdateMain.java

```
public class UpdateMain {
    public static void main(String[] args) {
        try {
            String cfgFileName = "mybatis_config.xml";
            InputStream cfgIs = Resources.getResourceAsStream(cfgFileName);
            SqlSessionFactory ssf = new SqlSessionFactoryBuilder().build(cfgIs);
            SqlSession ss = ssf.openSession();

            Scanner sc = new Scanner(System.in);
            System.out.print("ID : ");
            String p_id = sc.next();
            System.out.print("가격 : ");
            BigDecimal p_price = sc.nextBigDecimal();

            Product p = new Product(p_id, null, p_price);

            int row = ss.update("kwon.updateProduct", p);

            if(row == 1){
                System.out.println("update 성공");
            }
            ss.commit();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```





4.3.3 실행결과

ID : mer
가격 : 999999
update 성공

p01 컴퓨터 800000
p02 마우스 12000
p03 보드마카 1000
p04 쇼핑백 2000
p05 usb케이블 2000
p06 우산 5000
p08 의자 100000
p10 모니터 200000
mt 물통 300
gkb 키보드 10000
mer 메르스소독제 999999
rb1 르블랑 500
fbk 노란꽃가방 100000





4.4 DELETE

4.4.1 mapper.xml 에 추가

```
<delete id="deleteProduct" parameterType="com.kwon.Product">
    delete from jun11_product_kwon
    where p_id=#{p_id}
</delete>
```

4.4.2 DeleteMain.java

```
public class DeleteMain {
    public static void main(String[] args) {
        try {
            String cfgFileName = "mybatis_config.xml";
            InputStream cfgIs = Resources.getResourceAsStream(cfgFileName);
            SqlSessionFactory ssf = new SqlSessionFactoryBuilder().build(cfgIs);
            SqlSession ss = ssf.openSession();

            Scanner sc = new Scanner(System.in);
            System.out.print("ID : ");
            String p_id = sc.next();

            Product p = new Product(p_id, null, null);

            int row = ss.update("kwon.deleteProduct", p);

            if (row == 1) {
                System.out.println("delete 성공");
            }
            ss.commit();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```





4.4.3 실행결과

ID : **rbl**
delete 성공

p01 컴퓨터 800000
p02 마우스 12000
p03 보드마카 1000
p04 쇼핑백 2000
p05 usb케이블 2000
p06 우산 5000
p08 의자 100000
p10 모니터 200000
mt 물통 300
gkb 키보드 10000
mer 메르스소독제 999999
fbk 노란꽃가방 100000

