

## MyBatis (一)

笔记本: mybatis

创建时间: 2018/12/23 16:13

更新时间: 2019/3/30 20:51

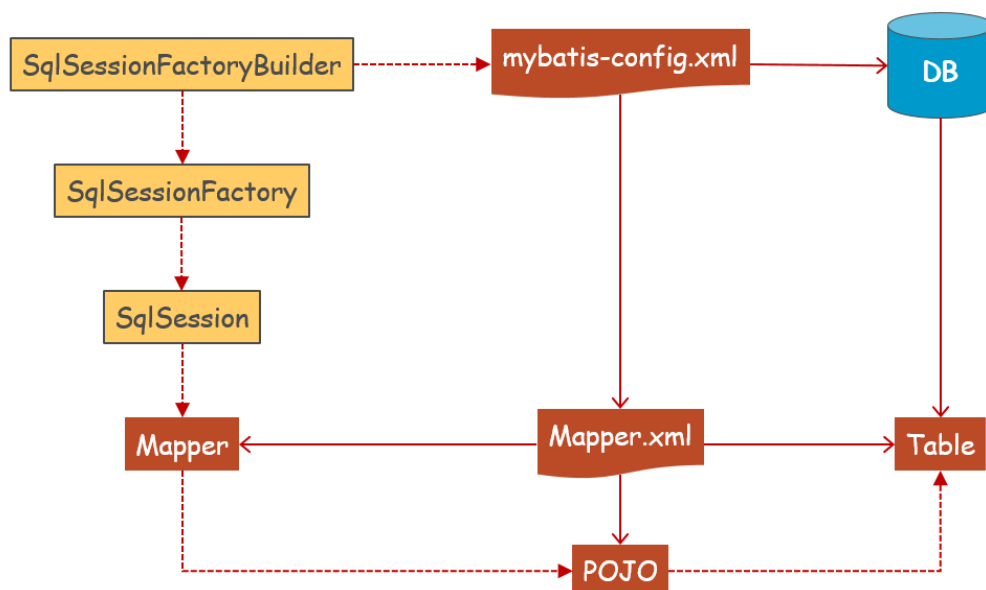
作者: qmys2018@163.com

<http://www.mybatis.org/mybatis-3/zh/index.html>

### MyBatis概述

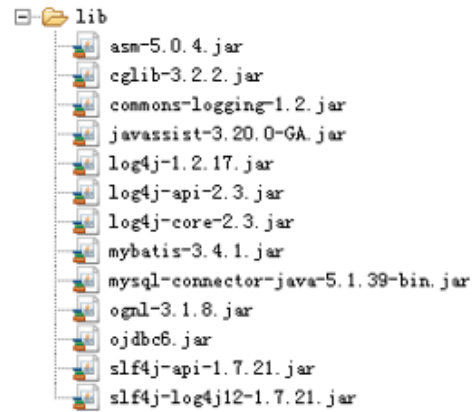
- MyBatis是支持定制化SQL、存储过程以及高级映射的优秀的持久层框架。
- MyBatis避免了几乎所有的JDBC代码和手动设置参数以及获取结果集。
- MyBatis可以对配置和原生Map使用简单的XML或注解，将接口和Java的POJO（Plain Old Java Object，普通的Java对象）映射成数据库中的记录。

### MyBatis体系结构



### 第一个MyBatis程序

- **User.java**
- **UserMapper.java**
- **UserMapper.xml**
- **mybatis-config.xml**
- **log4j.properties**
- **Demo0101.java**
- **Demo0102.java**
- **Demo0103.java**
- **Demo0104.java**



```
create table user_(
  id_ int primary key auto_increment,
  name_ varchar(50) not null
);
```

## User.java

```
package model;
public class User {
    private Integer id;
    private String name;
    public Integer getId() {
        return id;
    }
    public void setId(Integer id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}
```

## UserMapper.java

```
package mapper;
public interface UserMapper {
    User selectByPrimaryKey(Integer id);
    int insert(User record);
    int updateByPrimaryKey(User record);
    int deleteByPrimaryKey(Integer id);
}
```

## UserMapper.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="mapper.UserMapper">
    <resultMap id="BaseResultMap" type="model.User">
        <id column="ID_" property="id" jdbcType="DECIMAL" />
        <result column="NAME_" property="name" jdbcType="VARCHAR" />
    </resultMap>
    <select id="selectByPrimaryKey" resultMap="BaseResultMap"
parameterType="java.lang.Integer">
        select ID_, NAME_ from USER_ where ID_ = #{id,jdbcType=DECIMAL}
    </select>
    <insert id="insert" parameterType="model.User">
        insert into USER_ (ID_, NAME_)
        values (#{id,jdbcType=DECIMAL}, #{name,jdbcType=VARCHAR})
    </insert>
    <update id="updateByPrimaryKey" parameterType="model.User">
        update USER_ set NAME_ = #{name,jdbcType=VARCHAR}
        where ID_ = #{id,jdbcType=DECIMAL}
    </update>
    <delete id="deleteByPrimaryKey" parameterType="java.lang.Integer">
        delete from USER_ where ID_ = #{id,jdbcType=DECIMAL}
    </delete>
</mapper>

```

## 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="mysql">
        <environment id="mysql">
            <transactionManager type="JDBC" />
            <dataSource type="POOLED">
                <property name="driver" value="com.mysql.jdbc.Driver" />
                <property name="url" value="jdbc:mysql://localhost:3306/mydb" />
                <property name="username" value="root" />
                <property name="password" value="123456" />
            </dataSource>
        </environment>
    </environments>
    <mappers>
        <mapper resource="mapper/mysql/UserMapper.xml" />
    </mappers>
</configuration>

```

## log4j.properties

```

log4j.rootLogger=debug,stdout,logfile
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
#log4j.appender.stdout.Target=System.err
log4j.appender.stdout.layout=org.apache.log4j.SimpleLayout
log4j.appender.logfile=org.apache.log4j.FileAppender
log4j.appender.logfile.File=jbit.log
log4j.appender.logfile.layout=org.apache.log4j.PatternLayout
log4j.appender.logfile.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %F %p
%m%n
log4j.logger.com.ibatis=DEBUG
log4j.logger.com.ibatis.common.jdbc.SimpleDataSource=DEBUG
log4j.logger.com.ibatis.common.jdbc.ScriptRunner=DEBUG
log4j.logger.com.ibatis.sqlmap.engine.impl.SqlMapClientDelegate=DEBUG
log4j.logger.java.sql.Connection=DEBUG
log4j.logger.java.sql.Statement=DEBUGlog4j.logger.java.sql.PreparedStatement=DEBUG

```

## Demo0101.java

```
InputStream in = Resources.getResourceAsStream("mybatis-config.xml");
SqlSessionFactory factory = new SqlSessionFactoryBuilder().build(in);
SqlSession session = factory.openSession();
UserMapper userMapper = session.getMapper(UserMapper.class);
User user = new User();
user.setId(1);
user.setName("Alex");
userMapper.insert(user);
session.commit();
session.close();
```

## 结果集

### ■ POJO

- Mapper
  - POJO get(Integer id);
  - List<POJO> find();
- Mapper.xml
  - <select resultMap="">
    - <resultMap id="" type="POJO">
  - <select resultType="POJO">
    - 要求字段名与属性名一致

### ■ Map

- Mapper
  - Map get(Integer id);
  - List<Map> find();
- Mapper.xml
  - <select resultType="map">

## UserMapper1.java

```
package mapper;
public interface UserMapper1 {
    User get(Integer id);
    List<User> find();
    Map<String, Object> getMap(Integer id);
    List<Map<String, Object>> findMap();
}
```

## UserMapper1.xml

```
<mapper namespace="mapper.UserMapper1">
    <resultMap id="BaseResultMap" type="model.User">
        <id column="ID_" property="id" jdbcType="DECIMAL" />
        <result column="NAME_" property="name" jdbcType="VARCHAR" />
    </resultMap>
```

```

<select id="get" resultMap="BaseResultMap">
    select ID_, NAME_ from USER_ where ID_ = #{id}
</select>
<select id="find" resultType="model.User">
    select ID_ id, NAME_ name from USER_
</select>
<select id="getMap" resultType="map">
    select ID_, NAME_ from USER_ where ID_ = #{id}
</select>
<select id="findMap" resultType="map">
    select ID_, NAME_ from USER_
</select>
</mapper>

```

## Demo0105.java

```

InputStream in = Resources.getResourceAsStream("mybatis-config.xml");
SqlSessionFactory factory = new SqlSessionFactoryBuilder().build(in);
SqlSession session = factory.openSession();
UserMapper1 userMapper1 = session.getMapper(UserMapper1.class);
User user = userMapper1.get(1);
System.out.println(user.getName());
List<User> list = userMapper1.find();
System.out.println(list.size());
session.close();

```

## Demo0106.java

```

InputStream in = Resources.getResourceAsStream("mybatis-config.xml");
SqlSessionFactory factory = new SqlSessionFactoryBuilder().build(in);
SqlSession session = factory.openSession();
UserMapper1 userMapper1 = session.getMapper(UserMapper1.class);
Map<String, Object> map = userMapper1.getMap(1);
System.out.println(map.get("NAME_"));
List<Map<String, Object>> list = userMapper1.findMap();
System.out.println(list.size());
session.close();

```

## 参数

### ■ 单个参数

- Mapper
  - void insert(POJO);
  - void insert(Map);
- Mapper.xml
  - #{属性名}
  - #{关键字}

### ■ 多个参数

- Mapper
  - void insert(int p1,String p2);
  - void insert(@Param("p1") int p1,@Param("p2") String p2);
- Mapper.xml

- #{位置}
- #{名称}

## UserMapper2.java

```
package mapper;
public interface UserMapper2 {
    void insert1(Map<String, Object> map);
    void insert2(Integer id, String name);
    void insert3(@Param("id") Integer id, @Param("name") String name);
}
```

## UserMapper2.xml

```
<mapper namespace="dao.UserMapper2">
    <insert id="insert1">
        insert into USER_ (ID_, NAME_)
        values (#{id,jdbcType=DECIMAL}, #{name,jdbcType=VARCHAR})
    </insert>
    <insert id="insert2">
        insert into USER_ (ID_, NAME_)
        values (#{0,jdbcType=DECIMAL}, #{1,jdbcType=VARCHAR})
    </insert>
    <insert id="insert3">
        insert into USER_ (ID_, NAME_)
        values (#{id,jdbcType=DECIMAL}, #{name,jdbcType=VARCHAR})
    </insert>
</mapper>
```

## Demo0107.java

```
InputStream in = Resources.getResourceAsStream("mybatis-config.xml");
SqlSessionFactory factory = new SqlSessionFactoryBuilder().build(in);
SqlSession session = factory.openSession();
UserMapper2 userMapper2 = session.getMapper(UserMapper2.class);
Map<String, Object> map = new HashMap<String, Object>();
map.put("id", 2);
map.put("name", "Amy");
userMapper2.insert1(map);
userMapper2.insert2(3, "Angela");
userMapper2.insert3(4, "Alice");
session.commit();
session.close();
```

## 自动生成主键

■支持自动生成主键的JDBC驱动

- <insert useGeneratedKeys="true" keyProperty="">
- </insert>

## UserMapper3.xml

```
<mapper namespace="dao.UserMapper3">
  <insert id="insert" useGeneratedKeys="true" keyProperty="id">
    insert into USER_ (NAME_) values (#{name,jdbcType=VARCHAR})
  </insert>
</mapper>
```

## 总结

### ■MyBatis概述

### ■MyBatis体系结构

### ■第一个MyBatis程序

### ■MyBatis核心组件

### ■结果集

### ■参数

### ■自动生成主键