

阿进的写字台

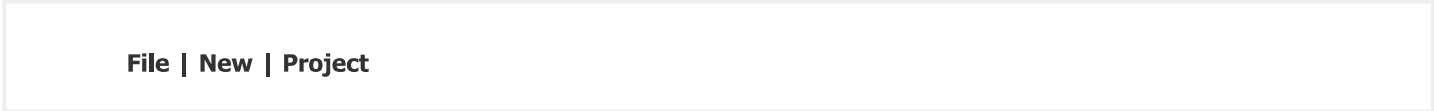
博客园 首页 新随笔 联系 订阅 管理

随笔 - 49 文章 - 0 评论 - 54

mybatis 初步使用（IDEA的Maven项目, 超详细）

创建 Maven 项目

1. 菜单上选择新建项目



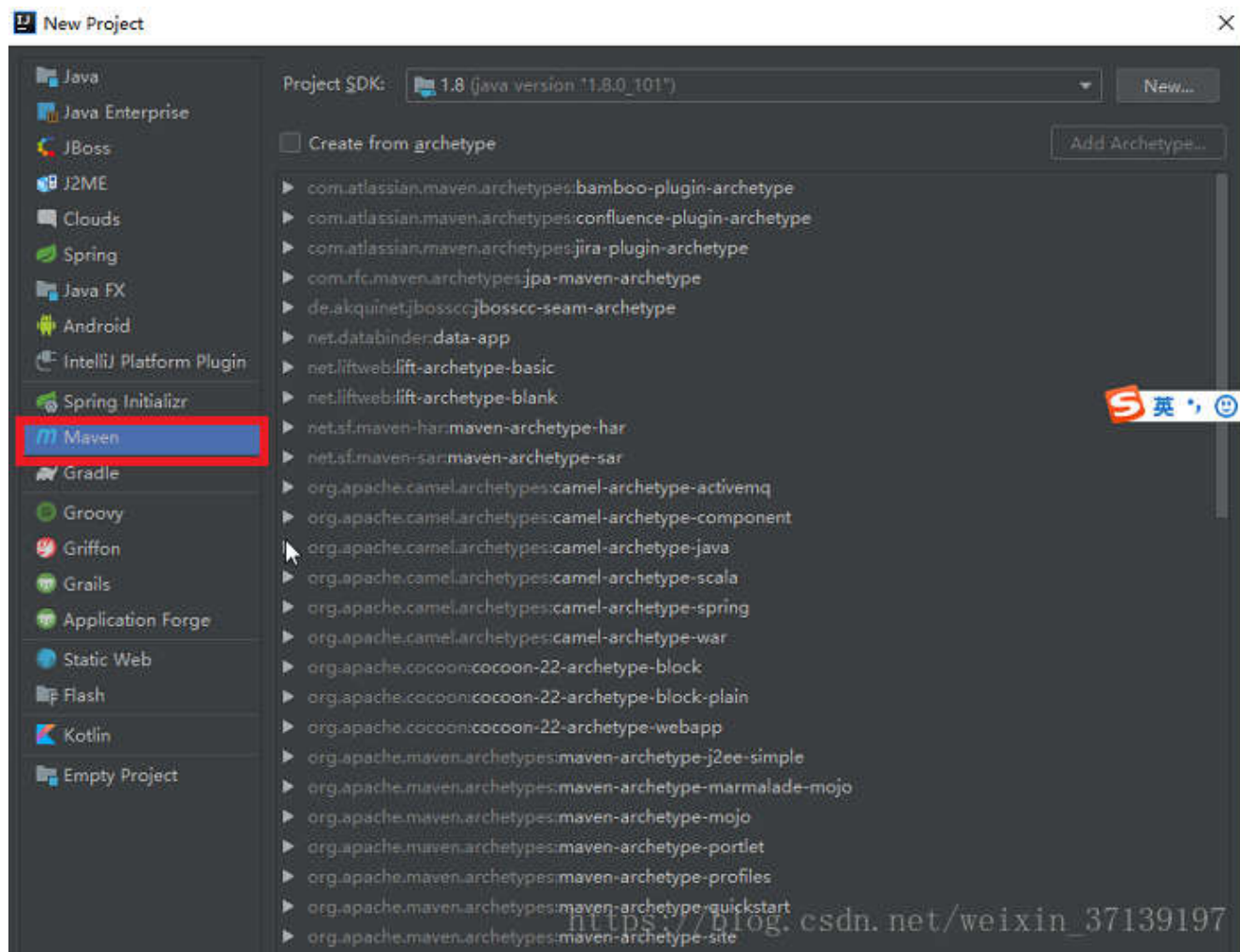
2. 选择左侧的 Maven

昵称：阿进的写字台  
园龄：1年6个月  
粉丝：47  
关注：3  
+加关注

2018年12月						
日	一	二	三	四	五	六
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

搜索

- 常用链接
- [我的随笔](#)
- [我的评论](#)
- [我的参与](#)
- [最新评论](#)



由于我们只是创建一个普通的项目, 此处点击 **Next** 即可。

### 3. 输入 `GroupId` 和 `ArtifactId`

- 在我的项目中,

GroupId 填 com.homejim.mybatis

ArtifactId 填 helloMybatis

#### 我的标签

##### 我的标签

Java(26)

mybatis(17)

mybatis使用(7)

mybatis源码(6)

iot(6)

mqtt(6)

paho(6)

Thread(5)

多线程(4)

反射(3)

更多

##### 随笔分类

C/C++(1)

Clean Code

Java(8)

Java 进阶(15)

kafka(1)

linux(1)

mqtt(6)

mybatis-工具(1)

点击 Next 。

4. Finish

通过以上步骤，一个普通的 Maven 项目就创建好了。

## Maven配置

为了让 mybatis 在项目上跑起来，需要进行如下配置。

### pom.xml

#### 1. 配置 JDK 版本和编码方式

```
<plugins>
  <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>2.3.2</version>
    <configuration>
      <source>1.8</source>
      <target>1.8</target>
      <encoding>UTF-8</encoding>
    </configuration>
  </plugin>
</plugins>
```

设置后，编码方式为 UTF-8，JDK 版本为 1.8

#### 2. 设置资源文件路径

Maven 中默认是只会打包 resource 下的资源文件。如果我们的文件不放在 resource，则需要通过配置告知 Maven。

```
<resources>
  <resource>
    <directory>src/main/java</directory>
    <includes>
      <include>**/*.properties</include>
      <include>**/*.xml</include>
```

mybatis-使用(8)

mybatis-源码(10)

paho(5)

spring-boot使用

tomcat(1)

多线程那些事(1)

集合框架那些事(1)

### 随笔档案

2018年12月 (5)

2018年11月 (5)

2018年10月 (11)

2018年9月 (8)

2018年8月 (5)

2018年3月 (3)

2018年1月 (3)

2017年12月 (9)

### linux

### 积分与排名

积分 - 29332

排名 - 17422

### 最新评论

```
        </includes>
        <filtering>false</filtering>
    </resource>
</resources>
```

## 2. 添加 mybatis 依赖

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

本项目的 `mybatis` 版本为 `3.4.5`。

## 3. 添加数据库驱动依赖

```
<!--数据库 mysql 驱动-->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.45</version>
</dependency>
```

本项目使用的是 `mysql` 数据库。

## 4. 添加日志依赖

```
<!-- 实现slf4j接口并整合 -->
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>1.7.25</version>
</dependency>
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-log4j12</artifactId>
    <version>1.7.25</version>
</dependency>
```

1. Re:MyBatis动态SQL（认真看看，以后写SQL就爽多了）

@灯塔下的守望者得到肯定，很开心，谢谢...

--阿进的写字台

2. Re:mybatis源码- 反射模块一（跟着MyBatis学反射）：类级别信息的封装

@Java深入浅出这个判断很简单的。if ((name.startsWith("get") && name.length() > 3) || (name.startsW.....

--阿进的写字台

3. Re:MyBatis动态SQL（认真看看，以后写SQL就爽多了）

支持，非常好

--灯塔下的守望者

4. Re:mybatis源码- 反射模块一（跟着MyBatis学反射）：类级别信息的封装

怎么没写怎么判断 setter和getter?

--Java深入浅出

5. Re:HashMap是如何工作的

mark

--Java深入浅出

### 阅读排行榜

1. Paho -物联网 MQTT C Client的实现和详解(9888)

```
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-core</artifactId>
  <version>2.10.0</version>
</dependency>
```

添加日志是为了在测试时可以输出 `sql` 语句。

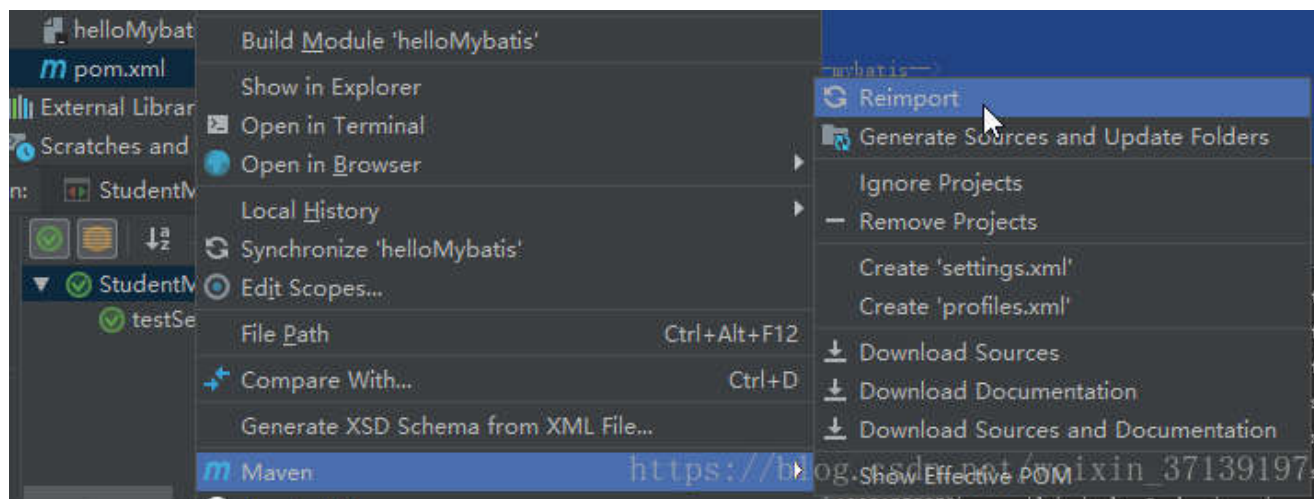
## 5. 添加测试依赖

```
<!--junit 测试-->
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>4.12</version>
  <scope>test</scope>
</dependency>
```

添加 `junit` 是为了测试更加方便。

## 6. 导入依赖

右键 `pom.xml` | **Maven** | **Reimport**



2. 运行时动态库: not found 及介绍-linux的-Wl,-rpath命令(3506)

3. JDBC详解系列 (一) 之流程(3348)

4. Paho - MQTT C Client的实现(3031)

5. MyBatis动态SQL (认真看看, 以后写SQL就爽多了) (2198)

## 评论排行榜

1. MyBatis动态SQL (认真看看, 以后写SQL就爽多了) (14)

2. mybatis 多个接口参数的注解使用方式 (@Param)(7)

3. mybatis源码-解析配置文件 (二) 之解析的流程(3)

4. mybatis源码-原来resultMap解析完是这样(3)

5. Java设计模式-建造者(Builder)模式(2)

## 推荐排行榜

1. MyBatis动态SQL (认真看看, 以后写SQL就爽多了) (15)

2. HashMap是如何工作的(5)

3. 扒一扒: Java 中的枚举(4)

4. mybatis源码-原来resultMap解析完是这样(4)

5. mybatis源码-Mapper解析之SQL 语句节点解析 (一条语句对应一个MappedStatement)(2)

## 创建数据库

### 1. 创建数据库

```
CREATE DATABASE IF NOT EXISTS `mybatis` DEFAULT CHARSET utf8 COLLATE utf8_general_ci;
```

指定编码为

### 2. 创建数据库表

创建一个  表。

```
DROP TABLE IF EXISTS `student`;  
CREATE TABLE `student` (  
  `student_id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '编号',  
  `name` VARCHAR(20) DEFAULT NULL COMMENT '姓名',  
  `phone` VARCHAR(20) DEFAULT NULL COMMENT '电话',  
  `email` VARCHAR(50) DEFAULT NULL COMMENT '邮箱',  
  `sex` TINYINT(4) DEFAULT NULL COMMENT '性别',  
  `locked` TINYINT(4) DEFAULT NULL COMMENT '状态(0:正常,1:锁定)',  
  `gmt_created` DATETIME DEFAULT CURRENT_TIMESTAMP COMMENT '存入数据库的时间',  
  `gmt_modified` DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '修改的时间',  
  PRIMARY KEY (`student_id`)  
) ENGINE=INNODB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4 COMMENT='学生表';
```

### 3. 插入测试数据

插入一些测试数据。

```
INSERT INTO `student`(`name`, phone, email, sex, locked)  
VALUES  
( '小明', 13821378270, 'xiaoming@mybatis.cn', 1, 0),  
( '小丽', 13821378271, 'xiaoli@mybatis.cn', 0, 0),  
( '小刚', 13821378272, 'xiaogang@mybatis.cn', 1, 0),  
( '小花', 13821378273, 'xiaohua@mybatis.cn', 0, 0),  
( '小强', 13821378274, 'xiaoqiang@mybatis.cn', 1, 0),  
( '小红', 13821378275, 'xiaohong@mybatis.cn', 0, 0);
```

# 配置Mybatis

## 配置mybatis的XML文件

XML 配置文件（configuration XML）中包含了对 MyBatis 系统的核心设置。

### 1. 复制框架

从[官网](#)中复制最基本的配置：

```
<?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="${driver}"/>
        <property name="url" value="${url}"/>
        <property name="username" value="${username}"/>
        <property name="password" value="${password}"/>
      </dataSource>
    </environment>
  </environments>
  <mappers>
    <mapper resource="org/mybatis/example/BlogMapper.xml"/>
  </mappers>
</configuration>
```

### 2. 配置 environments

`environments` 对应的是数据库相关的属性。我们可以配置多个环境，但每个 `SqlSessionFactory` 实例只能选择其一。在本项目中，我们只配置一个即可。数据库时刚刚我们创建的 `mybatis`。

```
<environments default="development">
  <environment id="development">
    <transactionManager type="JDBC">
      <property name="" value=""/>
    </transactionManager>
  </environment>
</environments>
```

```
</transactionManager>
<dataSource type="POOLED">
  <property name="driver" value="com.mysql.jdbc.Driver"/>
  <property name="url" value="jdbc:mysql://localhost:3306/mybatis"/>
  <property name="username" value="root"/>
  <property name="password" value="你自己的密码"/>
</dataSource>
</environment>
</environments>
```

### 3. 配置 settings

在本项目中， 主要是为了配置日志输出。 `pom.xml` 中配置的是 `log4j`， 因此此处的日志实现值是 `LOG4J`。

```
<settings>
  <setting name="logImpl" value="LOG4J" />
</settings>
```

### 4. 配置 typeAliases

类型别名是为 Java 类型设置一个短的名字。它只和 XML 配置有关， 存在的意义仅在于用来减少类完全限定名的冗余。

如果我们这样配置

```
<!--项目中不是这样配置的-->
<typeAliases>
  <typeAlias alias="Student" type="com.homejim.mybatis.entity.Student"/>
</typeAliases>
```

那么在 `mybatis` 中， `Student` 可以用在任何使用 `com.homejim.mybatis.entity.Student` 的地方。

但一般来说 `entity` 都会有很多， 因此这样配置太繁琐了， `mybatis` 支持指定包名， 会使用类的首字母小写的类限定名作为别名。

本项目中， 如下配置

```
<typeAliases>
  <package name="com.homejim.mybatis.entity"/>
</typeAliases>
```

### 5. 配置 mappers



同理，可以单个配置，也可以通过扫描包来识别。

```
<mappers>
  <package name="com.homejim.mybatis.mapper"/>
</mappers>
```

该配置会扫描指定包中的所有 `.xml` 文件，这些文件包含着 `mybatis` 的 SQL 语句和映射配置文件。后续会创建。

## 创建实体类和对应的Mapper.xml

### 1. 创建实体类

实体类，属性需要与数据库表中的字段一一对应，并有对应的 `getter` 和 `setter`。

```
package com.homejim.mybatis.entity;

import java.util.Date;

public class Student {
    private Integer studentId;

    private String name;

    private String phone;

    private String email;

    private Byte sex;

    private Byte locked;

    private Date gmtCreated;

    private Date gmtModified;
    /**
     * 以下部分为setter和getter，省略
     */
}
```

### 2. 创建 Mapper 接口和 Mapper.xml

StudentMapper.java

```
package com.homejim.mybatis.mapper;

import com.homejim.mybatis.entity.Student;

import java.util.List;

public interface StudentMapper {

    /**
     *
     * @return
     */
    List<Student> selectAll();
}
```

StudentMapper.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="com.homejim.mybatis.mapper.StudentMapper">
    <resultMap id="BaseResultMap" type="com.homejim.mybatis.entity.Student">
        <id column="student_id" jdbcType="INTEGER" property="studentId" />
        <result column="name" jdbcType="VARCHAR" property="name" />
        <result column="phone" jdbcType="VARCHAR" property="phone" />
        <result column="email" jdbcType="VARCHAR" property="email" />
        <result column="sex" jdbcType="TINYINT" property="sex" />
        <result column="locked" jdbcType="TINYINT" property="locked" />
        <result column="gmt_created" jdbcType="TIMESTAMP" property="gmtCreated" />
        <result column="gmt_modified" jdbcType="TIMESTAMP" property="gmtModified" />
    </resultMap>
    <sql id="Base_Column_List">
        student_id, name, phone, email, sex, locked, gmt_created, gmt_modified
    </sql>
    <select id="selectAll" resultMap="BaseResultMap">
        select
        <include refid="Base_Column_List" />
    </select>
</mapper>
```

```
from student
</select>
</mapper>
```

配置了结果映射和 `sql` 语句, 其中 `sql` 中的 `id` 与 `Mapper` 中的方法名一致。

## 测试

```
public class StudentMapperTest {
    private static SqlSessionFactory sqlSessionFactory;

    @BeforeClass
    public static void init() {
        try {
            Reader reader = Resources.getResourceAsReader("mybatis-config.xml");
            sqlSessionFactory = new SqlSessionFactoryBuilder().build(reader);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    @Test
    public void testSelectList() {
        SqlSession sqlSession = null;
        try {
            sqlSession = sqlSessionFactory.openSession();

            List<Student> students = sqlSession.selectList("selectAll");
            for (int i = 0; i < students.size(); i++) {
                System.out.println(students.get(i));
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (sqlSession != null) {
                sqlSession.close();
            }
        }
    }
}
```

```
}  
}
```

1. 通过 `Resources` 工具类读取 `mybatis-config.xml` , 存入 `Reader` ;
2. `SqlSessionFactoryBuilder` 使用上一步获得的 `reader` 创建 `SqlSessionFactory` 对象;
3. 通过 `sqlSessionFactory` 对象获得 `SqlSession` ;
4. `SqlSession` 对象通过 `selectList` 方法找到对应的 “selectAll” 语句, 执行 `SQL` 查询。
5. 底层通过 `JDBC` 查询后获得 `ResultSet` , 对每一条记录, 根据 `resultMap` 的映射结果映射到 `Student` 中, 返回 `List` 。
6. 最后记得关闭 `SqlSession` 。

获得的结果如下

```
2018-09-04 23:35:16.926 [main] DEBUG [org.apache.ibatis.datasource.pooled.PooledDataSource] - Created connection 900008524.  
2018-09-04 23:35:16.926 [main] DEBUG [org.apache.ibatis.transaction.jdbc.JdbcTransaction] - Setting autocommit to false on JDBC Connection [com.mysql.jdbc.JDBC4Connection@35a50a4c]  
2018-09-04 23:35:16.926 [main] DEBUG [com.homejim.mybatis.mapper.StudentMapper.selectAll] - ==> Preparing: select student_id, name, phone, email, sex, locked, gmt_created, gmt_modified from student  
2018-09-04 23:35:17.004 [main] DEBUG [com.homejim.mybatis.mapper.StudentMapper.selectAll] - ==> Parameters:  
2018-09-04 23:35:17.051 [main] TRACE [com.homejim.mybatis.mapper.StudentMapper.selectAll] - <== Columns: student_id, name, phone, email, sex, locked, gmt_created, gmt_modified  
2018-09-04 23:35:17.051 [main] TRACE [com.homejim.mybatis.mapper.StudentMapper.selectAll] - <== Row: 1, 小明, 13821378270, xiaoming@mybatis.cn, 1, 0, 2018-09-04 10:07:52.0, 2018-09-04 10:07:52.0  
2018-09-04 23:35:17.067 [main] TRACE [com.homejim.mybatis.mapper.StudentMapper.selectAll] - <== Row: 2, 小丽, 13821378271, xiaoli@mybatis.cn, 0, 0, 2018-09-04 10:07:52.0, 2018-09-04 10:07:52.0  
2018-09-04 23:35:17.067 [main] TRACE [com.homejim.mybatis.mapper.StudentMapper.selectAll] - <== Row: 3, 小刚, 13821378272, xiaogang@mybatis.cn, 1, 0, 2018-09-04 10:07:52.0, 2018-09-04 10:07:52.0  
2018-09-04 23:35:17.067 [main] TRACE [com.homejim.mybatis.mapper.StudentMapper.selectAll] - <== Row: 4, 小花, 13821378273, xiaohua@mybatis.cn, 0, 0, 2018-09-04 10:07:52.0, 2018-09-04 10:07:52.0  
2018-09-04 23:35:17.067 [main] TRACE [com.homejim.mybatis.mapper.StudentMapper.selectAll] - <== Row: 5, 小强, 13821378274, xiaoliang@mybatis.cn, 1, 0, 2018-09-04 10:07:52.0, 2018-09-04 10:07:52.0  
2018-09-04 23:35:17.067 [main] TRACE [com.homejim.mybatis.mapper.StudentMapper.selectAll] - <== Row: 6, 小红, 13821378275, xiaohong@mybatis.cn, 0, 0, 2018-09-04 10:07:52.0, 2018-09-04 10:07:52.0  
2018-09-04 23:35:17.067 [main] DEBUG [com.homejim.mybatis.mapper.StudentMapper.selectAll] - <== Total: 6  
Student(studentId=1, name='小明', phone='13821378270', email='xiaoming@mybatis.cn', sex=1, locked=0, gmtCreated=Tue Sep 04 10:07:52 CST 2018, gmtModified=Tue Sep 04 10:07:52 CST 2018)  
Student(studentId=2, name='小丽', phone='13821378271', email='xiaoli@mybatis.cn', sex=0, locked=0, gmtCreated=Tue Sep 04 10:07:52 CST 2018, gmtModified=Tue Sep 04 10:07:52 CST 2018)  
Student(studentId=3, name='小刚', phone='13821378272', email='xiaogang@mybatis.cn', sex=1, locked=0, gmtCreated=Tue Sep 04 10:07:52 CST 2018, gmtModified=Tue Sep 04 10:07:52 CST 2018)  
Student(studentId=4, name='小花', phone='13821378273', email='xiaohua@mybatis.cn', sex=0, locked=0, gmtCreated=Tue Sep 04 10:07:52 CST 2018, gmtModified=Tue Sep 04 10:07:52 CST 2018)  
Student(studentId=5, name='小强', phone='13821378274', email='xiaoliang@mybatis.cn', sex=1, locked=0, gmtCreated=Tue Sep 04 10:07:52 CST 2018, gmtModified=Tue Sep 04 10:07:52 CST 2018)  
Student(studentId=6, name='小红', phone='13821378275', email='xiaohong@mybatis.cn', sex=0, locked=0, gmtCreated=Tue Sep 04 10:07:52 CST 2018, gmtModified=Tue Sep 04 10:07:52 CST 2018)
```

有SQL输出, 以及SQL的执行情况

## 源码

请访问我的 [Github](#) 主页获取。

作者: [阿进的写字台](#)

出处: <https://www.cnblogs.com/homejim/>

本文版权归作者和博客园共有, 欢迎转载, 但未经作者同意必须保留此段声明, 且在文章页面明显位置给出原文连接, 否则保留追究法律责任的权利。

分类: [mybatis-使用](#)

标签: [Java](#), [mybatis](#)

[好文要顶](#)[关注我](#)[收藏该文](#)

阿进的写字台

[关注 - 3](#)

[粉丝 - 47](#)

[+加关注](#)

1

0

« 上一篇: [静态代理和动态代理](#)

» 下一篇: [Java设计模式-建造者\(Builder\)模式](#)

posted @ 2018-09-09 11:56 阿进的写字台 阅读(349) 评论(2) 编辑 收藏

#1楼 2018-11-16 08:36 大浪淘沙2018

“

非常详细的说明

支持(0) 反对(0)”

#2楼[楼主] 2018-11-16 09:09 阿进的写字台

“

@ 大浪淘沙2018

谢谢夸奖，写了一系列关于 mybatis 的文章。

所有博客都是经过本人亲自验证的，有问题可以给我留言

支持(0) 反对(0)”

[刷新评论](#) [刷新页面](#) [返回顶部](#)

**注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。**

【推荐】超50万VC++源码: 大型组态工控、电力仿真CAD与GIS源码库!

【活动】华为云12.12会员节 云产品1折起 满额送Mate20 点击抢购

【推荐】服务器100%基准CPU性能，1核1G首年168元，限时特惠!



#### 相关博文：

- 超详细的Maven使用教程
- idea的maven项目下spring与mybatis整合
- 使用idea创建maven项目
- IDEA下使用maven构建web项目（SpringMVC+Mybatis整合）
- idea + mybatis generator + maven 插件使用

#### 最新新闻：

- 携程有忧，建章难退
  - 史玉柱重出江湖三年：属于他的时代，远去了
  - 苹果回应在德国被禁售：对判决结果很失望 将上诉
  - Facebook隐私问题再度发酵：议员呼吁国会介入
  - 网易严选首家线下店开业 可这是一门好生意吗？
- » 更多新闻...

Copyright ©2018 阿进的写字台