

## 03Mybatis使用注解映射器

### 一、创建数据库：

1. 在MySQL数据库中创建一个名称为mybatis\_learn的数据库

```
CREATE DATABASE mybatis_learn CHARSET=UTF8;
```

2. 选择创建的数据

```
USE mybatis_learn
```

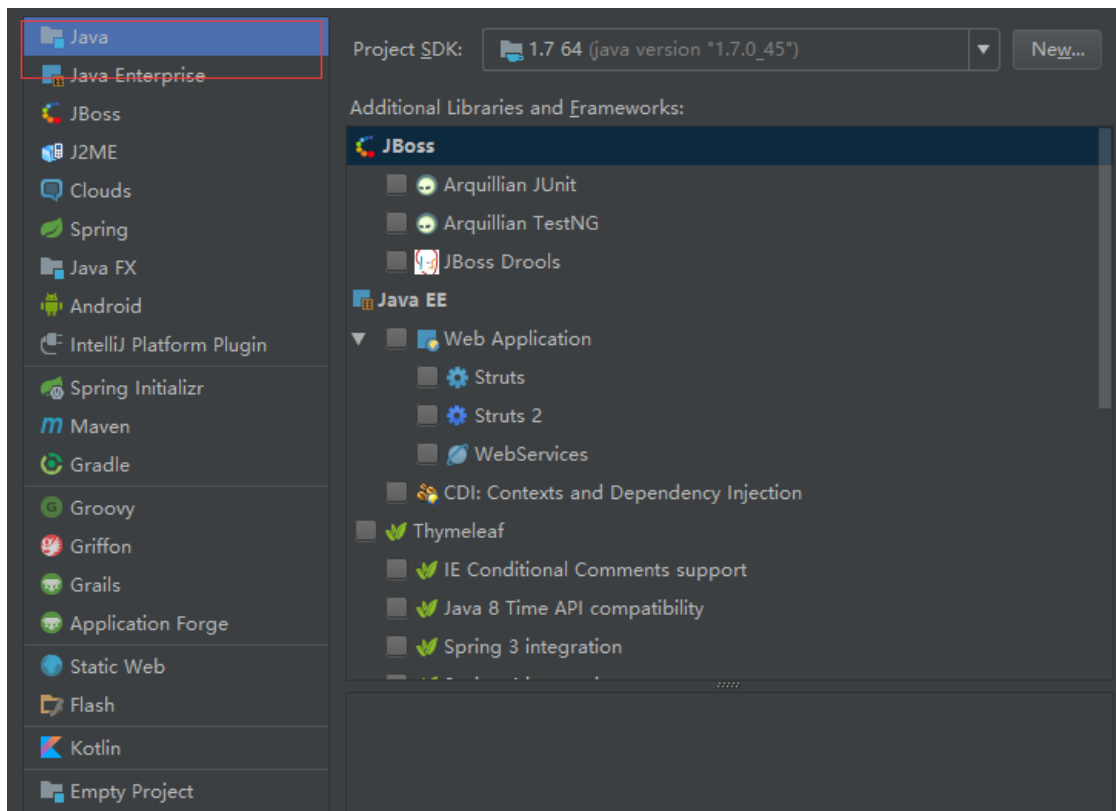
3. 创建表，并初始化数据

```
CREATE TABLE demo_user (  
  id int NOT NULL AUTO_INCREMENT,  
  user_id varchar(30) not null,  
  user_name varchar(30) not null,  
  nickname varchar(30),  
  password varchar(128) not null,  
  is_admin char(1) not null default 'N' comment 'N: 非管理员 Y: 管理员',  
  primary key (id)  
);  
  
INSERT INTO demo_user VALUES (1, '1', 'demo', 'demo', '123456a?', 'Y');  
INSERT INTO demo_user VALUES (2, '2', 'jack', 'jack', '123456', 'N');
```

### 二、创建java工程

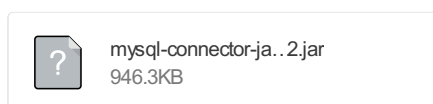
创建一个名字为mybatis\_learn的java工程文件

File-New-Project-Java然后next-next 填入要创建的工程名-finish



### 三、配置项目环境

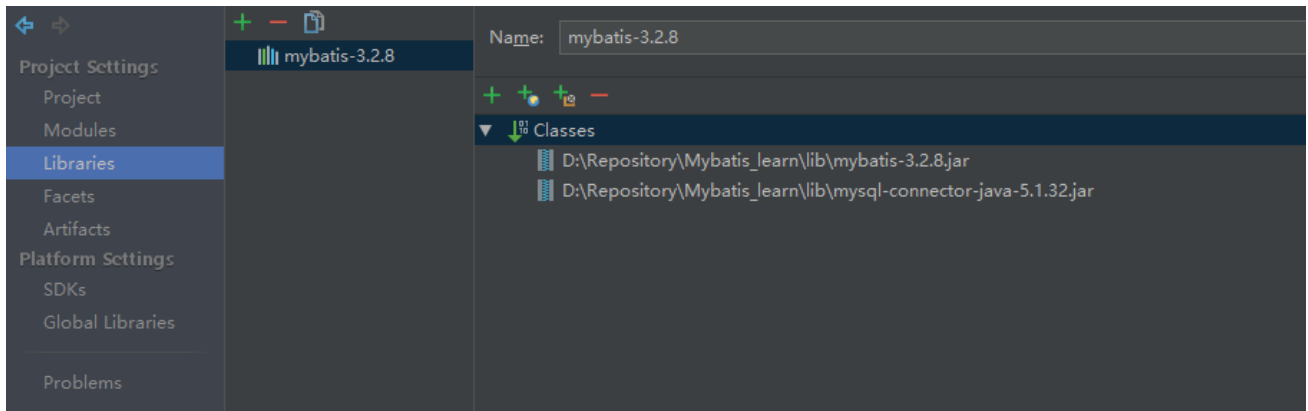
- 1.在此工程下创建一个lib目录，将所需依赖包导入





mybatis-3.2.8.jar  
697.33KB

## 2. 右击项目，点击Open model Settings选择Libraries然后添加所需jar包



## 3. 添加datasource.properties配置文件

此配置文件用来存放链接数据库的信息

- 右击src目录new一个file文件输入文件名：datasource.properties
- 向datasource.properties文件中添加配置信息如下所示

```
dataSource.driverClassName=com.mysql.jdbc.Driver  
dataSource.url=jdbc:mysql://localhost:3306/mybatis_learn?useUnicode=true&characterEncoding=utf-8  
dataSource.username=root  
dataSource.password=inspur
```

说明：

mybatis\_learn 为要连接的数据库的名字  
root 为登录数据库的用户名  
inspur为登录密码

## 4. 添加mybatis-config.xml配置文件

此配置文件用来存放mybatis的配置信息

方法跟上面一样，配置信息如下所示：

```
<?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>  
    <properties resource="datasource.properties"></properties>  
    <br />  
    <environments default="dataSource">  
        <environment id="dataSource">  
            <transactionManager type="JDBC" />  
            <dataSource type="POOLED">  
                <property name="driver" value="${dataSource.driverClassName}" />  
                <property name="url" value="${dataSource.url}" />  
                <property name="username" value="${dataSource.username}" />  
                <property name="password" value="${dataSource.password}" />  
            </dataSource>  
        </environment>  
    </environments>  
</configuration>
```

```
</environment>
</environments>
</configuration>
```

## 四、创建名称为UserMapper的接口

1. 在src下创建一个com.inspur.mybatis.demo.dao的包

右击src——new——package

2. 在此包下创建UserMapper接口

右击src——new——java选择接口

3. 在接口中编写getUserById() 查询方法

```
package com.inspur.mybatis.demo.dao;
import java.util.Map;
import org.apache.ibatis.annotations.Select;

public interface UserMapper {
    @Select("SELECT * FROM demo_user WHERE user_id = #{id}")
    public Map<String, Object> getUserById(int id);
}
```

4. 将UserMapper所在路径添加到mybatis的配置文件mybatis-config.xml中:

```
<?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>
    <properties resource="datasource.properties"></properties>
    <environments default="dataSource">
        <environment id="dataSource">
            <transactionManager type="JDBC" />
            <dataSource type="POOLED">
                <property name="driver" value="${dataSource.driverClassName}" />
                <property name="url" value="${dataSource.url}" />
                <property name="username" value="${dataSource.username}" />
                <property name="password" value="${dataSource.password}" />
            </dataSource>
        </environment>
    </environments>
    <mappers>
        <package name="com.inspur.mybatis.demo.dao" />
    </mappers>
</configuration>
```

## 5. 创建测试类

- a. 在src中创建com.inspur.mybatis.demo.test包

- b. 创建一个App类

```
package com.inspur.mybatis.demo.test;
import java.io.IOException;
import java.io.InputStream;
import java.util.Map;
import org.apache.ibatis.io.Resources;
import org.apache.ibatis.session.SqlSession;
import org.apache.ibatis.session.SqlSessionFactory;
```

```
import org.apache.ibatis.session.SqlSessionFactoryBuilder;
import com.inspur.mybatis.demo.dao.UserMapper;

public class App {

    public static void main(String[] args) {

        String resource = "mybatis-config.xml";
        SqlSession session = null;
        try {
            InputStream inputStream = Resources.getResourceAsStream(resource);
            SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(inputStream);
            session = sqlSessionFactory.openSession(true);
            UserMapper userMapper = session.getMapper(UserMapper.class);
            Map<String, Object> userMap = userMapper.getUserById(1);
            System.out.println(userMap);
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (session != null) {
                session.close();
            }
        }
    }
}
```