代码地址

https://github.com/lilewa/mybatisChicken/tree/master/start

面向对象编程出现多年，对象化在应用程序端已经得到很好的实践，但在流行的关系数据库端，数据并不是以对象方式组织。应用程序为了使用数据库的数据，不得不将从数据库得到的数据转为对象模式。ORM框架的出现，省去了转换数据这些繁重的工作，抹平了应用程序和数据库之间使用数据方式的差异。

MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。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 可以使用简单的 XML 或注解来配置和映射原生类型、接口和 Java 的 POJO为数据库中的记录。

与其他的ORM框架不同，MyBatis并没有将Java 对象与数据库表关联起来，而是将Java 方法与SQL 语句关联。

下面搭建一个简单的mybatis应用

**工具**

MySQL 8.0



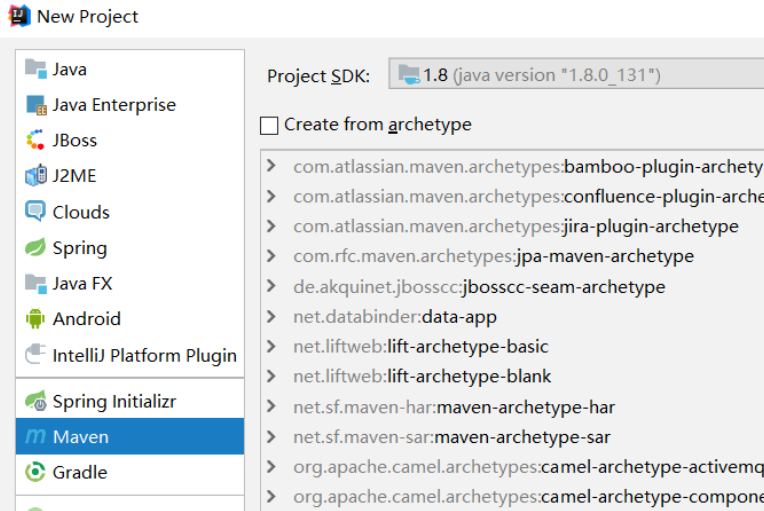
IntelliJ IDEA 2018.1.3 x64

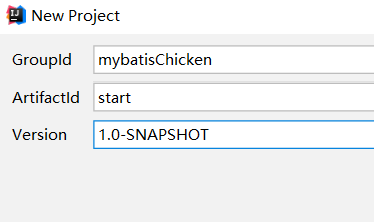


MySQL Workbench 8.0 CE



**新建一个maven项目**





**1.打开maven配置文件pom.xml，添加以下依赖项**

Mybatis

mysql-connector-java

log4j

junit

**2.准备数据库**

**2.1添加名为mybatis的数据库**

CREATE DATABASE mybatis DEFAULT CHARACTER SET utf8 COLLATE utf8\_general\_ci;

**2.2添加table**

CREATE TABLE `country` (  
 `id` int(11) NOT NULL AUTO\_INCREMENT,  
 `countryname` varchar(255) DEFAULT NULL,  
 `countrycode` varchar(255) DEFAULT NULL,  
 PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8;

**3.使用XML 配置mybatis**

MyBatis包含两种类型的xml文件，配置文件（config文件）和映射文件（mapper文件）。

config文件包含了影响 MyBatis 行为的设置和属性信息

mapper文件包含了配置的sql语句，存储过程以及映射规则

在src/main/resources 下面创建mybatis-config.xml 配置文件，并配置mybatis

**3.1配置日志输出**

在config文件中添加日志输出设置，logImpl属性名指定使用LOG4J 输出日志

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

**3.2配置数据库连接**

<dataSource type="UNPOOLED">  
 <property name="driver" value="com.mysql.cj.jdbc.Driver"/>  
 <property name="url" value="jdbc:mysql://localhost:3306/mybatis?serverTimezone=UTC"/>  
 <property name="username" value="root"/>  
 <property name="password" value="root"/>  
</dataSource>

**4.创建实体类**

src/main/java下添加一个包 mybatisChicken.start.model，在这个包下再新建实体类Country

public class Country {  
 private Long id;  
 private String countryname;  
 private String countrycode;

}

Country中的id使用类型Long而不是long。

由于Java 中的基本类型会有默认值，例如当某个类中存在private int age ；字段。创建这个类时， age 会有默认值0 。当使用age 字段时，它总会有值。因此在某些情况下，便无法实现使age 为null 。并且在动态SQL 的部分，如果使用age != null 进行判断，结果总会为true ，因而会导致很多隐藏的问题。所以，在实体类中不要使用基本类型。基本类型包括byte 、int , short 、long 、float 、double 、char 、boolean 。

**5.创建mapper文件**

src/main/resources下新建目录”mapper”，再在这个目录下新建CountryMapper文件。文件中mapper元素的namespace特性设置为noInterface.mapper.CountryMapper

<mapper namespace="noInterface.mapper.CountryMapper">  
 <select id="selectAll" resultType="mybatisChicken.start.model.Country">  
 select id,countryname,countrycode from country  
 </select>  
</mapper>

Namespace的值可以任意取，多个mapper文件可以共用一个namespace值，但同一namespace下不可以存在相同id的配置。上面配置了一个id为selectAll的查询语句。查询结果映射到resultType特性指定的类型上，mybatisChicken.start.model.Country是刚才创建的实体类

**6.回到config文件添加配置的mapper文件**

在mybatis-config.xml 配置文件的mappers元素中添加配置的mapper文件 。resource特性填入mapper文件的路径

<mappers>  
 <mapper resource="mapper/CountryMapper.xml"/>  
</mappers>

**7．配置log4j日志设置**

3.1设置了mybatis使用log4j管理日志输出，现在配置log4j

在 src/main/resources下添加log4j.properties文件，添加

log4j.logger.noInterface.mapper =TRACE

log4j.logger.noInterface.mapper对应的包是noInterface.mapper

在MyBatis 的日志实现中，包名实际上是XML 配置中的namespace 特性值的一部分。所以这里的配置要和mapper文件中namespace特性的值匹配

<mapper namespace="noInterface.mapper.CountryMapper">

**8．编写测试代码**

在src/test/java中创建包mybatisChicken.start，在这个包下创建测试类。CountryMapperTest

**8.1 实例化SqlSessionFactory**

SqlSessionFactory 是MyBatis的核心，通常一个数据库连接配置对应一个SqlSessionFactory实例。它的最佳作用域是应用作用域。最简单的做法是使用单例模式或者静态单例模式。

在@BeforeClass注解的方法中使用mybatis-config.xml配置文件初始化sqlSessionFactory实例

Reader reader = Resources.getResourceAsReader("mybatis-config.xml");  
sqlSessionFactory= new SqlSessionFactoryBuilder().build(reader);

**8.2在测试方法中获取数据**

SqlSession sqlSession = *sqlSessionFactory*.openSession();

List<Country> countryList = sqlSession.selectList( "noInterface.mapper.CountryMapper.selectAll" );

通过SqlSession 的selectList 方法使用namespace+id作为参数，在CountryMapper.xml中查找到要执行SQL 查询。MyBatis 底层使用JDBC 执行SQL ，获得查询结果集ResultSet 后，根据resultType特性的值将结果映射为Country 类型的集合， 返回查询结果。

**其他**

Mapper文件中每次指定类名时需使用类的完全限定名。在config文件中添加类的包名后可以简化类名的书写

<typeAliases>  
 <package name="mybatisChicken.start.model"/>  
</typeAliases>

Mapper文件中使用类名时，不需要再写类的包名。

<select id="selectAll" resultType="mybatisChicken.start.model.Country">简化为

<select id="selectAll" resultType=" Country">

引用

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

<https://github.com/mybatis-book/book>