狂神说MyBatis01：第一个程序

秦疆 狂神说 2020-04-07

狂神说MyBatis系列连载课程，通俗易懂，基于MyBatis3.5.2版本，欢迎各位狂粉转发关注学习，视频同步文档。

MyBatis简介

环境说明：

- jdk 8 +
- MySQL 5.7.19
- maven-3.6.1
- IDEA

学习前需要掌握：

- JDBC
- MySQL
- Java 基础
- Maven
- Junit

什么是MyBatis

- MyBatis 是一款优秀的**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的过程
- MyBatis 可以使用简单的 XML 或注解来配置和映射原生信息，将接口和 Java 的 实体类 【Plain Old Java Objects,普通的 Java对象】映射成数据库中的记录。
- MyBatis 本是apache的一个开源项目ibatis, 2010年这个项目由apache 迁移到了google code，并且改名为MyBatis 。

- 2013年11月迁移到**Github** .
- Mybatis官方文档 : http://www.mybatis.org/mybatis-3/zh/index.html
- GitHub : https://github.com/mybatis/mybatis-3

> 持久化

持久化是将程序数据在持久状态和瞬时状态间转换的机制。

- 即把数据（如内存中的对象）保存到可永久保存的存储设备中（如磁盘）。持久化的主要应用是将内存中的对象存储在数据库中，或者存储在磁盘文件中、XML数据文件中等等。
- JDBC就是一种持久化机制。文件IO也是一种持久化机制。
- 在生活中 : 将鲜肉冷藏，吃的时候再解冻的方法也是。将水果做成罐头的方法也是。

为什么需要持久化服务呢？那是由于内存本身的缺陷引起的

- 内存断电后数据会丢失，但有一些对象是无论如何都不能丢失的，比如银行账号等，遗憾的是，人们还无法保证内存永不掉电。
- 内存过于昂贵，与硬盘、光盘等外存相比，内存的价格要高2~3个数量级，而且维持成本也高，至少需要一直供电吧。所以即使对象不需要永久保存，也会因为内存的容量限制不能一直呆在内存中，需要持久化来缓存到外存。

> 持久层

什么是持久层？

- 完成持久化工作的代码块 . ----> dao层 【DAO (Data Access Object) 数据访问对象】
- 大多数情况下特别是企业级应用，数据持久化往往也就意味着将内存中的数据保存到磁盘上加以固化，而持久化的实现过程则大多通过各种**关系数据库**来完成。
- 不过这里有一个字需要特别强调，也就是所谓的“层”。对于应用系统而言，数据持久功能大多是必不可少的组成部分。也就是说，我们的系统中，已经天然的具备了“持久层”概念？也许是，但也许实际情况并非如此。之所以要独立出一个“持久层”的概念,而不是“持久模块”，“持久单元”，也就意味着，我们的系统架构中，应该有一个相对独立的逻辑层面，专注于数据持久化逻辑的实现.
- 与系统其他部分相对而言，这个层面应该具有一个较为清晰和严格的逻辑边界。【说白了就是用来操作数据库存在的！】

> 为什么需要Mybatis

- Mybatis就是帮助程序猿将数据存入数据库中 , 和从数据库中取数据 .
- 传统的jdbc操作 , 有很多重复代码块 .比如 : 数据取出时的封装 , 数据库的建立连接等等... , 通过框架可以减少重复代码,提高开发效率 .
- MyBatis 是一个半自动化的**ORM框架 (Object Relationship Mapping) -->对象关系映射**
- 所有的事情，不用Mybatis依旧可以做到，只是用了它，所有实现会更加简单！**技术没有高低之分，只有使用这个技术的人有高低之别**
- MyBatis的优点

- 简单易学：本身就很小且简单。没有任何第三方依赖，最简单安装只要两个jar文件+配置几个sql映射文件就可以了，易于学习，易于使用，通过文档和源代码，可以比较完全的掌握它的设计思路和实现。
- 灵活：mybatis不会对应用程序或者数据库的现有设计强加任何影响。sql写在xml里，便于统一管理和优化。通过sql语句可以满足操作数据库的所有需求。
- 解除sql与程序代码的耦合：通过提供DAO层，将业务逻辑和数据访问逻辑分离，使系统的设计更清晰，更易维护，更易单元测试。sql和代码的分离，提高了可维护性。
- 提供xml标签，支持编写动态sql。
- .....

- 最重要的一点，使用的人多！公司需要！

MyBatis第一个程序

思路流程：搭建环境-->导入Mybatis--->编写代码--->测试

> 代码演示

1、搭建实验数据库

```
CREATE DATABASE `mybatis`;

USE `mybatis`;

DROP TABLE IF EXISTS `user`;

CREATE TABLE `user` (
`id` int(20) NOT NULL,
`name` varchar(30) DEFAULT NULL,
`pwd` varchar(30) DEFAULT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

insert into `user`(`id`,`name`,`pwd`) values (1,'狂神','123456'),(2,'张三','abcdef'),(3,'李四','987654');
```

2、导入MyBatis相关 jar 包

- GitHub上找

```
<dependency>
   <groupId>org.mybatis</groupId>
   <artifactId>mybatis</artifactId>
   <version>3.5.2</version>
</dependency>
<dependency>
```

```
   <groupId>mysql</groupId>
   <artifactId>mysql-connector-java</artifactId>
   <version>5.1.47</version>
</dependency>
```

3、编写MyBatis核心配置文件

- 查看帮助文档

```
<?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="com.mysql.jdbc.Driver"/>
               <property name="url" value="jdbc:mysql://localhost:3306/mybatis?
useSSL=true&amp;useUnicode=true&amp;characterEncoding=utf8"/>
               <property name="username" value="root"/>
               <property name="password" value="123456"/>
           </dataSource>
       </environment>
   </environments>
   <mappers>
       <mapper resource="com/kuang/dao/userMapper.xml"/>
   </mappers>
</configuration>
```

4、编写MyBatis工具类

- 查看帮助文档

```
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 java.io.IOException;
import java.io.InputStream;

public class MybatisUtils {

   private static SqlSessionFactory sqlSessionFactory;

   static {
       try {
           String resource = "mybatis-config.xml";
           InputStream inputStream = Resources.getResourceAsStream(resource);
           sqlSessionFactory = new SqlSessionFactoryBuilder().build(inputStream);
       } catch (IOException e) {
```

```
            e.printStackTrace();
        }
    }

    //获取SqlSession连接
    public static SqlSession getSession(){
        return sqlSessionFactory.openSession();
    }

}
```

5、创建实体类

```
public class User {

    private int id;  //id
    private String name;   //姓名
    private String pwd;   //密码

    //构造,有参,无参
    //set/get
    //toString()

}
```

6、编写Mapper接口类

```
import com.kuang.pojo.User;
import java.util.List;

public interface UserMapper {
    List<User> selectUser();
}
```

7、编写Mapper.xml配置文件

- namespace 十分重要，不能写错！

```
<?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.kuang.dao.UserMapper">
 <select id="selectUser" resultType="com.kuang.pojo.User">
  select * from user
 </select>
</mapper>
```

8、编写测试类

- Junit 包测试

```
public class MyTest {
```

```
    @Test
    public void selectUser() {
        SqlSession session = MybatisUtils.getSession();
        //方法一:
        //List<User> users = session.selectList("com.kuang.mapper.UserMapper.selectUser");
        //方法二:
        UserMapper mapper = session.getMapper(UserMapper.class);
        List<User> users = mapper.selectUser();

        for (User user: users){
            System.out.println(user);
        }
        session.close();
    }
}
```

9、运行测试，成功的查询出来的我们的数据，ok！

> 问题说明

可能出现问题说明：Maven静态资源过滤问题

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

有了MyBatis以后再也不用写原生的JDBC代码了，舒服！

end

视频同步更新，这次一定！

“赠人玫瑰，手有余香”

狂神说 的赞赏码

狂神说