

# 第三周 Mybatis入门篇

## 1. mybatis是什么？

---

是一个轻量级的ORM框架，完全封装了JDBC代码，使得操作数据库变得简单，内部包含体系：

1. 基本CRUD操作
2. 缓存策略
3. 关联映射
4. 动态SQL支持

Mybatis作用于后端操作数据相关功能

## 2. Mybatis的基本使用步骤

---

1. 添加依赖

<!--添加mybatis依赖-->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>3.5.9</version>

</dependency>

<!-- 添加mysql驱动依赖-->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.49</version>

</dependency>



## 2. 创建核心配置文件(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>

<!-- 环境配置:事务配置,数据源配置-->

<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/hrms?"

serverTimezone=UTC"/>



```
        <property name="username"
value="root"/>
        <property name="password"
value="123456"/>
    </dataSource>
</environment>
</environments>
<!--设置映射接口所在的包-->
<mappers>
    <package name="com.softem.mapper"/>
</mappers>
</configuration>
```

### 3. 创建实体类和映射接口

- 实体类创建

```
public class Emp {

    private long eno;
    private String ename;
    private String sex;
    private String job;
    private java.sql.Date hiredate;
    private long dno;

    //Constructor
    //setter/getter
    //toString
}
```



- 映射接口

```
public interface EmpMapper {
```



```
    @Select("select * from emp")
```

```
    List<Emp> selectAll();
```

```
    @Insert("insert into emp(ename, sex, job, hiredate, dno) values(#{ename},#{sex},#{job},#{hiredate},#{dno})")
```

```
    boolean insert(Emp emp);
```

```
    @Delete("delete from emp where eno=#{eno}")
```

```
    boolean deleteByEno(long eno);
```

```
}
```

#### 4. 测试

```
public class TestMybatis {
```



```
    public static void main(String[] args) throws IOException {
```

```
        // 从 XML 中构建 SqlSessionFactory
```

```
        String resource = "mybatis-config.xml";
```

```
        InputStream inputStream =
```

```
Resources.getResourceAsStream(resource);
```

```
        SqlSessionFactory sqlSessionFactory = new  
SqlSessionFactoryBuilder().build(inputStream);
```

```
        //获取sqlsession
```

```
        SqlSession session =
```

```
sqlSessionFactory.openSession(true);
```

```
        //获取Mapper对象(反射, 设计模式之代理模式)
```

```
        EmpMapper mapper =
```

```
session.getMapper(EmpMapper.class);
```

```

        Emp e = new Emp();
        e.setEname("德玛西亚");
        e.setJob("坦克");
        e.setSex("男");
        e.setHiredate(new
Date(System.currentTimeMillis()));
        e.setDno(40);
        mapper.insert(e);

        List<Emp> list = mapper.selectAll();
        list.forEach(System.out::println);
    }
}

```

### 3. 使用映射文件配置映射关系

Mybatis映射的配置包含两种常见方式：

1. 基于注解的配置
2. **基于映射文件的配置**

```

<?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.softem.mybatis.mapper.UserMap
per">

    <insert id="insert">

```



```
        insert into tbuser(username,
password, phone, status) VALUES
        ({username},{password},{phone},{
status})
    </insert>

    <delete id="deleteById">
        delete from tbuser where id=#{id}
    </delete>

    <update id="update">
        update tbuser set password=#
{password} where id=#{id}
    </update>

    <select id="selectAll"
resultType="com.softem.mybatis.entity.User">
        select * from tbuser
    </select>

    <select id="selectById"
resultType="com.softem.mybatis.entity.User">
        select * from tbuser where id=#{id}
    </select>

</mapper>
```

## 注意事项

1. `namespace` 需要指定为映射接口的完全限定名
2. 每一个数据操作功能需要指定 `id`，且 `id` 属性值必须为接口中方法名

3. 对于查询操作必须指定 `resultType` 或者 `resultMap`
4. 映射文件配置完成后必须在核心配置文件中注册

```
<?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>  
    <!-- 环境配置:略-->  
  
    <!-- 设置映射接口所在的包-->  
    <mappers>  
        <mapper  
            resource="mappers/UserMapper.xml" />  
    </mappers>  
</configuration>
```



## 4. Idea的Mybatis插件使用

---

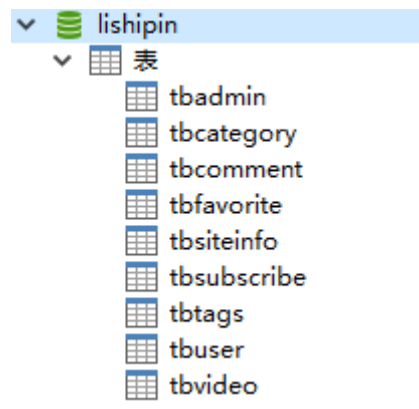
## 5. 练习

---

访问阿里云数据库:

`rm-bp1slbe3bhurcpfe23o.mysql.rds.aliyuncs.com`

提供了一个实例 `lishipin`，要求完成该实例中 `tbvideo` 表的CRUD操作



```
boolean insert(Video video)
```

```
boolean update(Video video)
```

```
boolean delete(int id)
```

```
Video selectById(int id)
```

```
List<Video> selectAll()
```

