

mybatis--offcn

- 1.持久存储
- 2.orm框架---技术选型
- 3.mybatis特点:
- 4.实体类注意点
- 5.mybatis事务
- 6.mybatis配置注意项
- 7.模糊查询
- 8.mapper文件编译bug点
- 9.注解学习方法
- 10.mybatis小知识细节弥补
- 11.返回主键值
- 12.sql片段重用
- 13.动态sql
- 14.全局文件mybatis-config.xml细节解析
- 15.多表联合查询注意点
- 16.mybatis的懒加载和缓存
- 17.mybatis的注解开发
- 18.mybatis的分页查询插件

1.持久存储

- 文件，网络，数据库

2.orm框架---技术选型

- orm: 如何使用面向对象操作数据库crud
- mybatis: java数据类型到数据库类型的映射
- Hibernate: 妥妥的orm框架，不写sql

Hibernate:
ORM框架==>不写SQL==>"表都不用创建"

MyBatis:
需要编写SQL语句 (MySQL 优化)
类和表 类属性与表字段映射
通过操作java对象实现对数据库的CRUD

- 技术选型的区别就出来了

3.mybatis特点:

- 面向接口编程, 持久层, orm映射, 三层架构怎么回事.
- 为什么要使用接口:

mybatis, 如果不适用面向接口的话, 我们在正常业务层调用持久层的时候, 会使用如下的代码, 就会造成, java在运行字节码文件都需要编译, 如果dao层文件改了, service层也需要重新编译, 耦合度很高, 使用xml之后, 替代了dao层实现类, 无需编译, 这service层也就不需要编译, 做到了解耦

```
Java | 复制代码

1 public class UserService{
2     private UserDao userDao=new UserDaoImpl();
3     方法一(){
4         userDao.方法1();
5     }
6 }
```

4.实体类注意点

- implements Serializable 缓存必须序列化
- 基本数据类型建议使用包装类: 原因基本类型会有默认值
- * 这种通配符就不要写了, 写字段, 优化效率高
- 注意我们在工具sql中写好, 写完粘贴过来,防止出错

5.mybatis事务

- mybatis默认开启了事务管理, 增删改都需要事务管理,
- opensession(true) 会让事务自动提交
- 将以用原生的默认方式, 该提交提交, 回滚的回滚
- 事务提交代码一定在service中

6.mybatis配置注意项

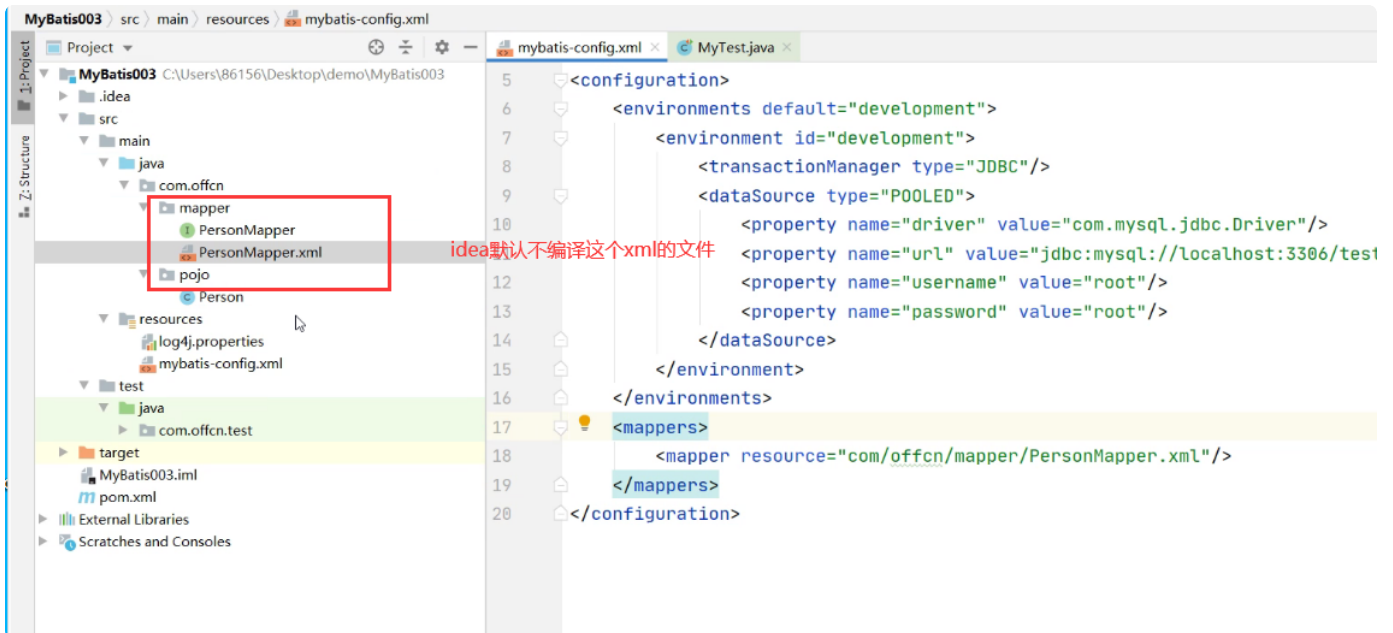
8.0 : com.mysql.cj.jdbc.driver

5.0 : com.mysql.jdbc.driver

7.模糊查询

- #{name}-----用户不会传%张%
- "%#{name}%"-----Oracle数据库除了起别名可以使用双引号，别的都不行
- '%\${name}%'-----sql注入有问题
- concat(concat('%',#{name}),'%')-----嵌套拼接是为了兼容Oracle数据库

8.mapper文件编译bug点



- 解决方案

```

</dependencies>
<build>
  <resources>
    <resource>
      <directory>src/main/java</directory>
      <includes>
        <include>**/*.xml</include>
      </includes>
    </resource>
  </resources>
</build>
</project>

```

9. 注解学习方法

- 注解位置
- 注解单词
- 注解在不同位置表示的意义

10. mybatis 小知识细节弥补

- 别名映射有两种，别名不区分大小写

```

<!-- 配置别名 -->
<!-- <typeAliases>-->
<!-- <typeAlias type="com.offcn.pojo.Person" alias="person"></typeAlias>-->
<!-- </typeAliases>-->
<typeAliases>
  <package name="com.offcn.pojo"/>
</typeAliases>

```

- 错误提示关键字---让你怎么改

提示可能需要干啥
Available parameters are [arg1, arg0, param1, param2]

- 参数映射



XML | 复制代码

```
1  <!--参数映射    -->
2  <!--只映射不同字段也可以-->
3  <resultMap id="selectUserMap" type="user">
4      <id column="uid" property="uid"></id>
5      <result column="phone" property="phone"></result>
6  </resultMap>
7  <select id="selectUser" resultMap="selectUserMap">
8      select * from user
9  </select>
```

11.返回主键值

XML | 复制代码

```
1  <!--主键回填-->指的是向数据库添加了一条数据需要立即返回它的主键（不走查
   询）-->
2  <insert id="insertUser" parameterType="user" useGeneratedKeys="true"
   keyProperty="uid">
3      insert into user (name,phone)values (#{name},#{phone})
4      <!--只支持那种自动增长的主键，但是oracle不支持-->
5  </insert>
6  <insert id="insertUser" parameterType="user">
7      <selectKey keyProperty="uid" resultType="Integer" order="AFTER">
8          select last_insert_id()
9      </selectKey>
10     insert into user (name,phone)values (#{name},#{phone})
11 </insert>
```

12.sql片段重用

```

1  <sql id="baseSql">
2      name, phone
3  </sql>
4  <select id="selectUser" resultMap="selectUserMap">
5      select <include refid="baseSql"></include> from user
6  </select>

```

13. 动态sql

- `<where></where>` 查询
- `<set></set>` 修改
- `<choose></choose>` 只会命中一个条件，条件一旦命中，后面不在看
- `<trim></trim>` set where
- `<foreach></foreach>` map, 数组, 集合, 对象---collection 写的是 map, array, list, 对象属性名称 xml 中不认识 & 符号, 要使用 &

14. 全局文件 mybatis-config.xml 细节解析

- properties 属性



```

<configuration>
  <!-- 自定义属性集 -->
  <!-- <properties> -->
  <!-- <property name="devUrl" value="jdbc:mysql://localhost:3306/java1129?useUnicode=true&characterEncoding=utf8" /> -->
  <!-- <property name="driverClass" value="com.mysql.jdbc.Driver" /> -->
  <!-- </properties> -->
  <!-- 引用外部配置文件 -->
  <properties resource="db.properties"></properties>

```

自定义属性集可以在后面标签引入

- typeAliases

```

<!-- 起别名: 忽略大小写 -->
<typeAliases>
  <!-- 为某一种类型起别名 -->
  <!-- <typeAlias type="com.offcn.pojo.Person" alias="person"></typeAlias> -->
  <!-- 为某一包下所有的类型起别名: 别名就是类名 -->
  <package name="com.offcn.pojo" />
</typeAliases>

```

- mappers 标签

```
<!-- 使用相对于类路径的资源引用 -->
<mappers>
  <mapper resource="org/mybatis/builder/AuthorMapper.xml"/>
  <mapper resource="org/mybatis/builder/BlogMapper.xml"/>
  <mapper resource="org/mybatis/builder/PostMapper.xml"/>
</mappers>
```

```
<!-- 使用完全限定资源定位符 (URL) -->
<mappers>
  <mapper url="file:///var/mappers/AuthorMapper.xml"/>
  <mapper url="file:///var/mappers/BlogMapper.xml"/>
  <mapper url="file:///var/mappers/PostMapper.xml"/>
</mappers>
```

```
<!-- 使用映射器接口实现类的完全限定类名 -->
<mappers>
  <mapper class="org.mybatis.builder.AuthorMapper"/>
  <mapper class="org.mybatis.builder.BlogMapper"/>
  <mapper class="org.mybatis.builder.PostMapper"/>
</mappers>
```

```
<!-- 将包内的映射器接口实现全部注册为映射器 -->
<mappers>
  <package name="org.mybatis.builder"/>
</mappers>
```

既有找mapper.xml又要找
mapper接口

所以层级目录相同

接口和mapper.xml位置

15.多表联合查询注意点

1. 一对多,一对一, 多对多都是一样的

- 主表配置:注意select属性是指其他命名空间, 注意一定要加上javaType(一对一)和ofType(一对多), 不然会报错

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
3      "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
4  <mapper namespace="com.offcn.mapper.CourseMapper">
5      <resultMap id="BaseMap" type="Course">
6          <id column="cid" property="cid"></id>
7          <result column="courseName" property="courseName"></result>
8          <result column="descs" property="descs"></result>
9          <result column="courseType" property="courseType"></result>
10     </resultMap>
11     <!-- 联合映射 注意map的继承关系-->
12     <resultMap id="selectCourseAndDetailByCourseName3Map" type="Course"
13 extends="BaseMap">
14         <collection property="courseList" ofType="coursedetail">
15             <id column="id" property="id"/>
16             <result column="name" property="name"></result>
17             <result column="type" property="type"></result>
18             <result column="url" property="url"></result>
19         </collection>
20     </resultMap>
21     <select id="selectCourseAndDetailByCourseName3"
22 resultMap="selectCourseAndDetailByCourseName3Map">
23         select
24             c.cid,courseName,descs,courseType,id,name,type,url
25         from course c,coursedetail cd where c.cid=cd.cid and courseName=#{courseName}
26     </select>
27     <!-- 嵌套映射-->
28     <resultMap id="selectCourseByCourseNameMap" type="Course"
29 extends="BaseMap">
30         <collection
31             property="courseList"
32             ofType="com.offcn.bean.Coursedetail"
33             column="cid"
34
35             select="com.offcn.mapper.CoursedetailMapper.selectCourseDetailByCid">
36         </collection>
37     </resultMap>
38     <select id="selectCourseByCourseName"
39 resultMap="selectCourseByCourseNameMap">
40         select cid,courseName,descs,courseType from course where
41             courseName=#{courseName}
42     </select>
```


39 </mapper>

- 从表配置

▼ 注意select属性是命名空间地方

Java | 复制代码

```
1   <mapper namespace="com.offcn.mapper.CoursedetailMapper">
2       <resultMap id="BaseMapper" type="coursedetail">
3           <id column="id" property="id"/>
4           <result column="name" property="name"></result>
5           <result column="type" property="type"></result>
6           <result column="url" property="url"></result>
7       </resultMap>
8       <select id="selectCourseDetailByCid" resultMap="BaseMapper">
9           select id,name,type,url from coursedetail where cid= #{cid}
10      </select>
11  </mapper>
```

16.mybatis的懒加载和缓存

- 懒加载

针对于collection和association可以对其进行懒加载，也就是分布查询的时候一般才起作用；可以局部配置，和全局进行配置，局部配置，真对某一个sql语句查询生效，而全局的配置，针对所有生效。

1. 方法一：

```
<collection property="ordersList" ofType="com.offcn.pojo.Orders"
    column="c_id"
    select="com.offcn.mapper.OrdersMapper.queryByFk"
    fetchType="lazy">|
</collection>
```

局部延迟加载

2. 方法二

▼ 全局文件配置

Java | 复制代码

```
1   <settings>
2       <!-- 打印查询语句 -->
3       <setting name="logImpl" value="STDOUT_LOGGING" />
4       <setting name="lazyLoadingEnabled" value="true"/>
5   </settings>
```

注意：局部加载和全局加载可以同时配置，但遵守局部加载的规则，如果全局懒，某一个局部为立即，调用局以局部为主。

- 缓存

mybatis的缓存分为一级缓存和二级缓存，一级缓存是针对同一个sqlSession的，同一个会话有效，而二级缓存需要我们手动开启配置，二级缓存是不同的sqlSession访问同一个sql语句的时候的缓存。

配置

Java | 复制代码

```
1  全局配置中开启缓存可以使用
2  <setting name="cacheEnabled" value="true"/>
3
4  在mapper中使用缓存
5  <mapper namespace="cn.offcn.dao.EmployeeDao">
6  <!-- 开启二级缓存的支持 -->
7  <cache></cache>
8  </mapper>
```

17.mybatis的注解开发

注解开发

Java | 复制代码

```
1  @Select("select id,name,type,url,start_data,cid from coursedetail
   where cid=#{cid}")
2  @Results({
3      @Result(column = "id",property = "id"),
4      @Result(column = "name",property = "name"),
5      @Result(column = "type",property = "type"),
6      @Result(column = "start_data",property = "startData"),
7      @Result(column = "cid",property = "cid"),
8      @Result(property = "course"
9          ,javaType = Course.class
10         ,one = @One(select =
11             "com.offcn.bean.Course.CourseById")
12         ,column = "cid")
13     })
14  @Select @Delete @Update @Insert.....
```

18.mybatis的分页查询插件

- 使用手册：<https://pagehelper.github.io/docs/howtouse/>
- 调用图解

```
// 1.开启分页
PageHelper.startPage( pageNum: 1, pageSize: 5);
// 2.查询所有的数据
List<Person> personList = personMapper.queryAll();
// 3.分页处理数据
PageInfo<Person> pageInfo = new PageInfo<>(personList);
// 4.获取分页后的数据
List<Person> list = pageInfo.getList();
```