

Swxctx_解闻

RSS订阅

铭心刻骨，就要一意孤行；越是憧憬，越要风雨兼程；要走多远，才算走进森林_xc....

4

收藏

评论

微信

微博

QQ

个人资料

Swxctx

关注

原创438

粉丝62

喜欢28

评论54

等级：博客5

访问：29万+

积分：7121

排名：4017

勋章：



网店转让



联系我

点我发送邮件

点击扫码关注

最新文章

Golang-Gomail实现简单的邮件发送

Mac安装Mongodb详解

GO-Golang实现的简单爬虫

Go-一些简单的类型转换及处理

Go-实现程序零点定时启动

博主专栏

Java ee/se

阅读量：77289134 篇

C And C++-QT

阅读量：83608109 篇

Linux

阅读量：3565595 篇



Golang

归档

2018年4月	2篇
2018年3月	1篇
2017年12月	5篇
2017年8月	1篇
2017年7月	6篇

展开

最新评论

GO-Golang实现的简单爬虫
qq_36400145：大佬，有没有完整代码？

Web-网上在线支付
datoudatou823：页面输入不能跳转

C++关系运算符重载
qq_40427761：bool函数没看懂？这是怎么比较的

windows远程连接mysql ...
zdyjj1222：您好 按照您文中第二步，去掉TCP前面的注释后，数据库打不开了 您遇到这个问题了么？

Mybatis-Dao层开发之Ma...
changyinling520：非常棒！

JD.COM 京东



¥ 7399.00

1/6

联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
🗣 QQ客服 🗣 客服论坛

关于 招聘 广告服务 

©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

原

Mybatis-Dao层开发之Mapper接口

2016年12月01日 21:52:42

阅读量：10446

基本配置：详见 http://blog.csdn.net/qq_28796345/article/details/53402808

dao接口开发实现：详见 http://blog.csdn.net/qq_28796345/article/details/53427087

Mapper接口开发方法只需要程序员编写Mapper接口（相当于Dao接口），由Mybatis框架根据接口定义创建接口的动态代理对象，代理对象的方法体同上边Dao接口实现类方法。

Mapper接口开发需要遵循以下规范：

- 1、Mapper.xml文件中的namespace与mapper接口的类路径相同。
- 2、Mapper接口方法名和Mapper.xml中定义的每个statement的id相同
- 3、Mapper接口方法的输入参数类型和mapper.xml中定义的每个sql的parameterType的类型相同
- 4、Mapper接口方法的输出参数类型和mapper.xml中定义的每个sql的结果类型的类型相同

1、Mapper映射文件

定义mapper映射文件UserMapper.xml（内容同Users.xml），需要修改namespace的值为UserMapper接口路径。将UserMapper.xml放在classpath下mapper目录下。

```
[html]
1.  <?xml version="1.0" encoding="UTF-8" ?>
2.  <!DOCTYPE mapper
3.  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
4.  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
5.
6.  <!-- 命名空间，对sql进行分类化管理（sql隔离） -->
7.  <mapper namespace="com.sw.mapper.UserMapper">
8.      <!-- 在映射文件中配置sql语句 -->
9.      <!-- 通过select执行查询，id用于标识映射文件中的sql (Statement-id)
10.      将sql语句封装到mappedstatement中
11.      #{}表示占位符
12.      parameterType-指定输入参数的类型
13.      #{id}-id表示输入的参数，参数名称就是id,如如果输入参数是简单类型，#{ }中的参数可以任意
14.      resultType-指定sql输出结果所映射的java对象类型
15.      -->
16.      <!-- 通过id查询用户表的记录 -->
17.      <select id="findUserById" parameterType="int" resultType="com.sw.po.User">
18.          select *from user where id=#{id}
19.      </select>
20.
21.      <!-- 根据用户名模糊查询用户信息 -->
22.      <!-- resultType-指定单条记录所映射的对象类型
23.      ${ }拼接sql串，接收参数的内容，不加任何修饰，拼接在sql中 (存在sql漏洞)
24.      ${ }接收输入参数的内容，如果传入的类型是简单类型，${ }中只能使用value
25.      -->
26.      <select id="findUserByName" parameterType="java.lang.String" resultType="com.sw.po.User">
27.          SELECT *FROM USER WHERE username LIKE '%${value}%'
28.      </select>
29.
30.      <!-- 添加用户 -->
31.      <!-- 指定输入参数类型是pojo (包括用户信息)
32.      #{ }中指定pojo (User) 属性名,接收到pojo的属性值
33.      Mybatis通过OGNL获取对象的属性值
34.      -->
35.      <insert id="insertUser" parameterType="com.sw.po.User">
36.          <!-- 获取刚增加的记录主键
37.          返回id到pojo对象 (User)
38.          SELECT LAST_INSERT_ID():得到刚插入进去记录的主键值，只适用于自增逐主键
39.          keyProperty: 将查询到的主键值设置到parameterType指定的对象的id属性
40.          order:指SELECT LAST_INSERT_ID()的执行顺序，相对于insert来说 (before/after)
41.          resultType:指定SELECT LAST_INSERT_ID()的结果类型
42.          -->
43.          <selectKey keyProperty="id" order="AFTER" resultType="java.lang.Integer">
44.              SELECT LAST_INSERT_ID()
45.          </selectKey>
46.          INSERT INTO USER (username, birthday, sex, address) VALUES (#{username}, #{birthday}, #{sex}, #{address})
```

```

50.         首先通过uuid得到主键，然后将主键设置到id属性中
51.         其次在Inster执行的时候从User对象中取出id的属性值
52.         -->
53.         <!-- <select key keyProperty="id" order="BEFORE" resultType="java.lang.String">
54.             SELECT UUID()
55.         </selectKey>
56.         INSERT INTO USER (id,username,birthday,sex,address) VALUES (#{id},{username},{birthday},{sex},{address}) --
->
57.         </insert>
58.
59.         <!-- 根据id删除用户 -->
60.         <delete id="deleteUser" parameterType="java.lang.Integer">
61.             DELETE FROM USER WHERE id=#{id}
62.         </delete>
63.
64.         <!-- 根据id更新用户
65.             传入用户id以及相关更新信息
66.             #{id}:从输入的user对象中获取user的属性值
67.         -->
68.         <update id="updateUser" parameterType="com.sw.po.User">
69.             UPDATE USER SET username=#{username},birthday=#{birthday},sex=#{sex},address=#{address} WHERE id=#{id}
70.         </update>
71.     </mapper>

```

2、Mapper.java-接口文件

Mapper接口定义有如下特点：

- 1、Mapper接口方法名与Mapper.xml(UserMapper.xml)中定义的statement的id相同
- 2、Mapper接口方法的输入参数类型和mapper.xml (User.xml) 中定义的statement的parameterType的类型相同
- 3、Mapper接口方法的输出参数类型和mapper.xml中定义的statement的结果Type的类型相同

代码如下；

```

[java]
1. package com.sw.mapper;
2.
3. import java.util.List;
4.
5. import com.sw.po.User;
6.
7. /*
8.  * @Author swxctx
9.  * @time 2016年12月1日
10.  * @Explain:使用mapper接口（用户管理），相当于dao接口
11.  */
12. public interface UserMapper {
13.     //根据id查询用户的信息
14.     public User findUserById(int id) throws Exception;
15.     //根据用户名查看用户列表
16.     public List<User> findUserByName(String username) throws Exception;
17.     //添加用户
18.     public void insertUser(User user) throws Exception;
19.     //删除用户
20.     public void deleteUser(int id) throws Exception;
21. }

```

3、测试类

```

[java]
1. package com.sw.mapper.test;
2.
3. import java.io.InputStream;
4. import java.util.Date;
5. import java.util.List;
6.
7. import org.apache.ibatis.io.Resources;
8. import org.apache.ibatis.session.SqlSession;
9. import org.apache.ibatis.session.SqlSessionFactory;
10. import org.apache.ibatis.session.SqlSessionFactoryBuilder;
11. import org.junit.Before;
12. import org.junit.Test;
13.
14. import com.sw.mapper.UserMapper;

```

```

18.  *@Author swxctx
19.  *@time 2016年12月1日
20.  *@Explain:
21.  */
22.  public class UserMapperTest {
23.
24.      private SqlSessionFactory sqlSessionFactory;
25.
26.      @Before
27.      public void setUpBefore() throws Exception {
28.          //执行其他方法前需要创建SqlSessionFactory
29.          // mybatis配置文件
30.          String resource = "SqlMapConfig.xml";
31.          // 得到配置文件流
32.          InputStream inputStream = Resources.getResourceAsStream(resource);
33.
34.          // 创建会话工厂，传入mybatis的配置文件信息
35.          sqlSessionFactory = new SqlSessionFactoryBuilder()
36.              .build(inputStream);
37.
38.
39.      @Test
40.      public void testFindUserById() throws Exception{
41.          SqlSession sqlSession = sqlSessionFactory.openSession();
42.          //创建UserMapper对象，mybatis自动生成mapper代理对象
43.          UserMapper userMapper = sqlSession.getMapper(UserMapper.class);
44.          //调用UserMapper方法
45.          User user = userMapper.findUserById(27);
46.          sqlSession.close();
47.
48.          System.out.println(user);
49.      }
50.
51.      @Test
52.      public void testFindUserByName() throws Exception{
53.          SqlSession sqlSession = sqlSessionFactory.openSession();
54.          //创建UserMapper对象，mybatis自动生成mapper代理对象
55.          UserMapper userMapper = sqlSession.getMapper(UserMapper.class);
56.          //调用UserMapper方法
57.          List<User> user = userMapper.findUserByName("王");
58.
59.          sqlSession.close();
60.
61.          System.out.println(user);
62.      }
63.
64.      @Test
65.      public void testInsertUser() throws Exception{
66.          SqlSession sqlSession = sqlSessionFactory.openSession();
67.          //创建UserMapper
68.          UserMapper userMapper = sqlSession.getMapper(UserMapper.class);
69.          User user = new User();
70.          user.setUsername("李小二");
71.          user.setSex("1");
72.          user.setBirthday(new Date());
73.          user.setAddress("广州");
74.
75.          //调用方法
76.          userMapper.insertUser(user);
77.          //提交事务
78.          sqlSession.commit();
79.          //释放资源
80.          sqlSession.close();
81.      }
82.
83.      @Test
84.      public void testDeleteUser() throws Exception{
85.          SqlSession sqlSession = sqlSessionFactory.openSession();
86.          //创建UserMapper对象，mybatis自动生成mapper代理对象
87.          UserMapper userMapper = sqlSession.getMapper(UserMapper.class);
88.          //调用UserMapper方法
89.          userMapper.deleteUser(44);
90.          sqlSession.commit();
91.          sqlSession.close();
92.      }
93.  }

```

注：

selectOne和selectList

动态代理对象调用sqlSession.selectOne()和sqlSession.selectList()是根据mapper接口方法的返回值决定，如果返回list则调用selectList方法，如果返回单个对象则调用selectOne方法。

namespace

mybatis官方推荐使用mapper代理方法开发mapper接口，程序员不用编写mapper接口实现类，使用mapper代理方法时，输入参数可以使用pojo包装对象或map对象，保证dao的通用性。

版权声明：欢迎大家提出意见，指出错误，互相交流。 https://blog.csdn.net/qq_28796345/article/details/53427134

个人分类：[Java EE](#)

所属专栏：[Java ee/se](#)

查看更多>>

想对作者说点什么？

我来说两句



常银玲~Judy 2017-09-26 16:56:16 #1楼

非常棒！

上一页 1 下一页

通用Mapper视频教程

通用Mapper是一款非常好用的MyBatis插件，它能够帮助我们自动生成常用增删改查操作的SQL语句，克服MyBatis开发过程中需要大量编写SQL语句的弊端。由于通用Mapper是根据实体类的属性自动生成对应的SQL语句，所以可以说通用Mapper插件是对MyBatis在ORM方面不足的重要补充。

学院 2018年03月28日 16:38

MyBatis 通用 Mapper 实现原理

MyBatis 通用 Mapper 实现原理 文章出自 GitChat <http://gitbook.cn/books/59ed3c942f5a1d7161bad162/index.html...>



isea533 2017-11-09 20:42:55 阅读数：4312

MyBatis 实践 -Mapper与DAO - CSDN博客

MyBatis 实践 -Mapper与DAO标签: Java与存储MyBatis简介MyBatis前身是iBatis,是一个基于Java的数据持久层/对象关系映射(ORM)框架. MyBatis是对JDBC的封装,使...

2018-4-25

使用mybatis时,dao层mapper文件的基本示例 - CSDN博客

dao层mapper文件的示例 实体类 Brand: package cn.itcast.core.bean.product; import java.io.Serializable; public class Brand implements Serializable { private ...

2018-3-22

50万码农评论：英语对于程序员有多重要？

不背单词和语法，一个公式学好英语



Mybatis通用Mapper

极其方便的使用Mybatis单表的增删改查 项目地址:<http://git.oschina.net/free/MyMapper> 优点? 不客气的说,使用这个通用Mapper甚至能改变你对...

 isea533 2014-11-25 09:07:01 阅读数: 183677

MyBatis之整合Spring(Dao和Mapper两种方式) - CSDN博客

MyBatis 实践 -Mapper与DAO MyBatis 实践 -Mapper与DAO标签: Java与存储MyBatis简介MyBatis前身是iBatis,是一个基于Java的数据持久层/对象关系映射(ORM)框架....

2018-3-29

spring+mybatis通用dao层、service层的一些个人理解与..._CSDN博客

1、现在的绝大多数web应用,通常都以action、service、dao三层去组织代码,这样划分...mybatis sql in 查询 mapper与service层写法 当查询的参数只有一个时 findByIds...

2018-3-28

Mapper映射文件中的#{}与\${}

#{}实现的是向prepareStatement中的预处理语句中设置参数值, sql语句中#{表示一个占位符即?. select id="findUserById"p...

 ruhang 2017-08-18 18:12:36 阅读数: 719

Mybatis框架学习（二）—DAO开发方式和Mapper代理的开发方式

通过第一篇文章《Mybatis框架学习(一)—快速入门》我们基本上了解了mybatis是什么, mybatis的整体架构以及它解决了对于程序直接使用jdbc连接数据库所产生的问题。接下来讲下mybati...

 xnf1991 2016-06-22 22:32:45 阅读数: 8239

Mybatis总结(二):开发Dao的方式和Mapper的方式 - CSDN博客

MyBatis 实践 -Mapper与DAO MyBatis 实践 -Mapper与DAO标签: Java与存储MyBatis简介MyBatis前身是iBatis,是一个基于Java的数据持久层/对象关系映射(ORM)框架....

2018-4-18


mybatis学习之原始开发DAO与mapper代理开发DAO的区别 - CSDN博客

MyBatis 实践 -Mapper与DAO MyBatis 实践 -Mapper与DAO标签: Java与存储MyBatis简介MyBatis前身是iBatis,是一个基于Java的数据持久层/对象关系映射(ORM)框架...

2018-5-1

MyBatis中的@Mapper注解及配套注解使用详解（上）

前言: 从mybatis3.4.0开始加入了@Mapper注解, 目的就是为不再写mapper映射文件(那个xml写的是真的蛋疼。。。)。很恶心的一个事实是源码中并没有对于这个注解的详细解释 ...

 phenomenonsTell 2018-01-11 13:35:59 阅读数: 5070

优达学城机器学习 – 成就IT转型的你

硅谷一流技术大咖主讲, 为全球贡献3%的AI工程师。立即免费试听!



mapper.xml是怎样实现Dao层接口 - CSDN博客

既然我们的目的是用xml配置文件来实现dao层接口。那...(mapper.getInterfaceName()); //加载一个接口类 ...是一个基于Java的数据持久层/对象关系映射(ORM)框架...


2018-4-21

Mapper和Dao名字相同 - CSDN博客

2018-5-2

[MyBatis]mapperLocations属性通配符的使用

mapperLocations属性通配符的使用: 注意点: 在classpath后面的*必不可少, 缺少型号的话后面的通配符不起作用。 ...

 zhouzhiwengang 2016-03-08 11:17:36 阅读数: 1013

Mapper和Dao名字相同

<http://blog.csdn.net/zjf280441589/article/details/50760236>

 u013165110 2016-12-16 18:31:30 阅读数: 312

Mybatis Mapper动态代理方法 即 只写Dao接口 不写Dao的实现类

Mapper动态代理方法:程序员只需要写dao接口(Mapper),而不需要写dao实现类,由mybatis根据dao接口和映射文件中statement的定义生成接口实现代理对象。可以调用代理对象...

2018-4-19

undefined

关于mybatis的一个疑惑 (Mapper接口如何实例化的)

今天早上有个疑惑,在mybatis+SpringMVC的工程中,我只定义了一个接口UserMapper,而Spring就产生了一个UserMapper的对象,这个是我所不明白的,一个接口如何实例化。查...

 tanqidong1992 2015-08-27 13:40:21 阅读数: 10404

Mapper.xml理解

最开始对myBatis也没有很深的理解,最近写了很多mapper代码,也看了一下MyBatis指南 myBatis.xml 1. 按住ctrl键点击BackupPrinterMapper,就会进...

 u011393781 2016-09-20 15:50:41 阅读数: 925

让AutoMapper在你的项目里飞一会儿(转)

出处:<http://www.cnblogs.com/WeiGe/p/3835523.html> 先说说DTO DTO是个什么东东? DTO (Data Transfer Object) 就是数据传输对...

 CsethCRM 2016-10-26 14:54:32 阅读数: 11703

【MyBatis学习18】使用注解配置Mapper

本文博客地址: <http://blog.csdn.net/soonfly/article/details/67640653> (转载请注明出处) MyBatis支持使用注解来配置映射的sql语句,这...

 kingice1014 2017-04-20 15:57:49 阅读数: 6756

MyBatis 实践 -Mapper与DAO

MyBatis 实践 -Mapper与DAO标签: Java与存储MyBatis简介MyBatis前身是iBatis,是一个基于Java的数据持久层/对象关系映射(ORM)框架. MyBatis i...

 hanqing280441589 2016-02-28 15:34:23 阅读数: 21321

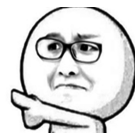
mybatis通用mapper的使用

项目中持久层封装了两套,一个hibernate,一个是mybatis. hibernate中封装了一些通用的方法,但是mybatis中没有,基于这个需求开始使用mybatis的通用mapper。 ...

 u013031185 2016-08-22 11:22:06 阅读数: 26313

50万码农评论: 英语对于程序员有多重要?

不背单词和语法,一个公式学好英语



 Smile_Miracle 2016-11-23 12:00:09 阅读数：3029

spring mybatis mapper接口注解方式注入

最近做项目使用mybatis-generator生成mybatis的mapper.xml映射文件和mapper接口。

 clj198606061111 2016-06-30 21:13:28 阅读数：44690

Java的MyBatis框架中Mapper映射配置的使用及原理解析

<http://www.jb51.net/article/85669.htm>

 u014401141 2016-07-27 14:43:05 阅读数：2364

java自动生成model和mapper的工具

2015年11月20日 33.3MB

下载



java中Jackson库的ObjectMapper类

ObjectMapper类是Jackson库的主要类。它提供一些功能将转换成Java对象匹配JSON结构，反之亦然。它使用JsonParser和JsonGenerator的实例实现JSON实际的读/写...

 lzq1125312193 2017-02-12 10:03:06 阅读数：2296

短信接口验证码

机器自动识别验证码的原理是怎么样的

百度广告



分离mybatis的mapper.java和mapper.xml文件

mapper.java和mapper.xml文件在不同的文件夹下

 u011277745 2017-02-20 21:07:24 阅读数：1621