

阿进的写字台

博客园 首页 新随笔 联系 订阅 管理

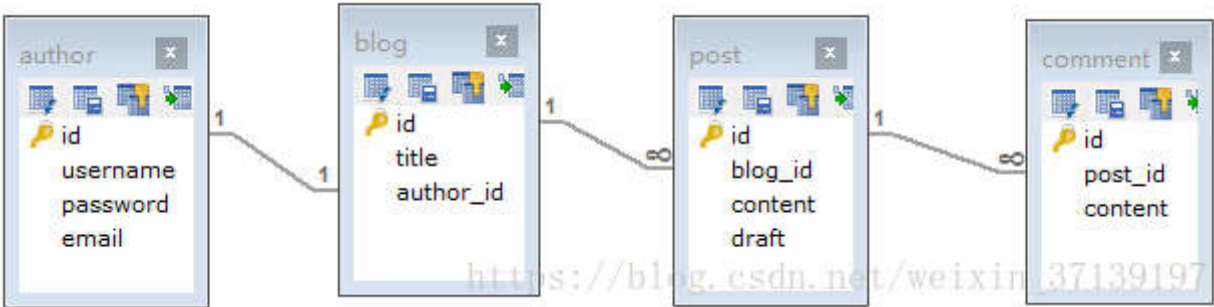
随笔 - 49 文章 - 0 评论 - 54

mybatis-高级结果映射之一对多

在一对多的关系中，主表的数据回对应关联表中的多条数据。因此，查询时就会查询出多条结果，所以，向类似的情况我们会使用 `List` 来进行存储关联表中获取到的信息。

1 数据准备

创建以下的名为 `mybatis` 的数据库，并在其下创建4个表。



在此就不贴出来建表的 SQL 语句了，感兴趣的可以去我的 [Github:mybatis-mapping](https://github.com/weixin37139197/mybatis-mapping) 中获取。

1.2 实体类，接口和XML

使用 [mybatis-代码生成器](#) 生成相应的实体类，接口和XML。

昵称：阿进的写字台
园龄：1年6个月
粉丝：47
关注：4
[+加关注](#)

2018年12月						
<	日	一	二	三	四	五 六
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

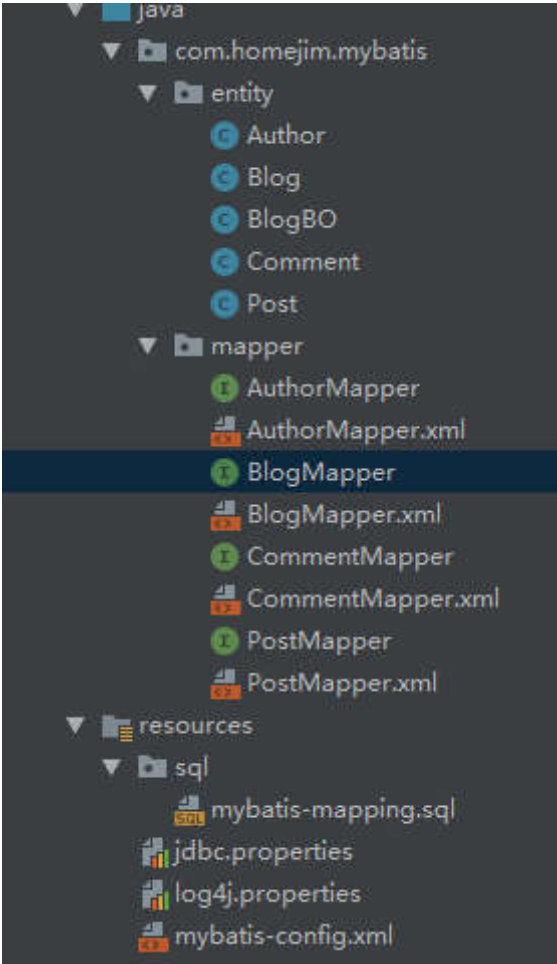
搜索

找找看

谷歌搜索

常用链接

- 我的随笔
- 我的评论
- 我的参与
- 最新评论



以上为生成的项目结构。

2 一对多映射

2.1 collection集合映射

2.1.1 创建结果实体类

我们需要创建一个 `BlogPostBO` ,

我的标签

我的标签

- Java(26)
- mybatis(17)
- mybatis使用(7)
- mybatis源码(6)
- iot(6)
- mqtt(6)
- paho(6)
- Thread(5)
- 多线程(4)
- 反射(3)
- 更多

随笔分类

- C/C++(1)
- Clean Code
- Java(8)
- Java 进阶(15)
- kafka(1)
- linux(1)
- mqtt(6)
- mybatis-工具(1)

```
public class BlogPostBO extends Blog {  
  
    private List<Post> posts;  
  
    public List<Post> getPosts() {  
        return posts;  
    }  
  
    public void setPosts(List<Post> posts) {  
        this.posts = posts;  
    }  
}
```

该类继承于 `Blog` 类，并多了一个链表成员变量 `posts`，我们后续获取到的发布的文章都在此链表中。

2.1.2 创建结果集

刚开始时，是这样子创建的。

```
<resultMap id="BlogPostBO" type="com.homejim.mybatis.entity.BlogPostBO" extends="BaseResultMap">  
    <collection property="posts" columnPrefix="post_" ofType="com.homejim.mybatis.entity.Post">  
        <id column="id" jdbcType="INTEGER" property="id" />  
        <result column="blog_id" jdbcType="INTEGER" property="blogId" />  
        <result column="draft" jdbcType="INTEGER" property="draft" />  
        <result column="content" jdbcType="LONGVARCHAR" property="content" />  
    </collection>  
</resultMap>
```

此处注意一个问题，在 **collection** 中，我们用的是 **ofType** 来表示 `List` 中的 `Pojo` 的属性。而不是 **type**。

因为我们内部 `Post` 有对应的结果集，可以引用另一个 `Mapper` 中的结果集，就可以简化变成下面这样子：

```
<resultMap id="BlogPostBO" type="com.homejim.mybatis.entity.BlogPostBO" extends="BaseResultMap">  
    <collection property="posts" columnPrefix="post_"  
        resultMap="com.homejim.mybatis.mapper.PostMapper.ResultMapWithBLOBs" />  
</resultMap>
```

可以简单了好多。

[mybatis-使用\(8\)](#)[mybatis-源码\(10\)](#)[paho\(5\)](#)[spring-boot使用](#)[tomcat\(1\)](#)[多线程那些事\(1\)](#)[集合框架那些事\(1\)](#)

随笔档案

[2018年12月 \(5\)](#)[2018年11月 \(5\)](#)[2018年10月 \(11\)](#)[2018年9月 \(8\)](#)[2018年8月 \(5\)](#)[2018年3月 \(3\)](#)[2018年1月 \(3\)](#)[2017年12月 \(9\)](#)

linux

积分与排名

积分 - 29332

排名 - 17422

最新评论

2.1.3 创建对应的方法和XML

首先就是在对应的 Mapper 接口下创建方法：

```
/**
 * 获取博客及其发布的内容 一对多
 * @return
 */
List<BlogPostBO> selectBlogAndPostList();
```

同时在 XML 中创建对应的 SQL 语句：

```
<select id="selectBlogAndPostList" resultMap="BlogPostBO">
  SELECT
    b.id,
    b.title,
    b.author_id,
    p.id as post_id,
    p.blog_id as post_blog_id,
    p.draft as post_draft,
    p.content as post_content
  FROM blog b left join post p on p.blog_id=b.id
</select>
```

2.1.4 测试

```
/**
 * resultMap + collection 一对多映射
 */
@Test
public void testSelectBlogAndPostListByBlog() {
    SqlSession sqlSession = sqlSessionFactory.openSession();
    BlogMapper blogMapper = sqlSession.getMapper(BlogMapper.class);
    List<BlogPostBO> blogPostBOs = blogMapper.selectBlogAndPostList();

    sqlSession.close();

    for (int i = 0; i < blogPostBOs.size(); i++) {
        BlogPostBO blogPostBO = blogPostBOs.get(i);
        System.out.println(blogPostBO.getTitle());
    }
}
```

1. Re:MyBatis动态SQL (认真看看, 以后写SQL就爽多了)

@灯塔下的守望者得到肯定, 很开心, 谢谢...

--阿进的写字台

2. Re:mybatis源码- 反射模块一 (跟着MyBatis学反射): 类级别信息的封装

@Java深入浅出这个判断很简单的。if ((name.startsWith("get") && name.length() > 3) || (name.startsW.....

--阿进的写字台

3. Re:MyBatis动态SQL (认真看看, 以后写SQL就爽多了)

支持, 非常好

--灯塔下的守望者

4. Re:mybatis源码- 反射模块一 (跟着MyBatis学反射): 类级别信息的封装

怎么没写怎么判断 setter和getter?

--Java深入浅出

5. Re:HashMap是如何工作的

mark

--Java深入浅出

阅读排行榜

1. Paho -物联网 MQTT C Client的实现和详解(9890)

```
for (int j = 0; j < blogPostBO.getPosts().size(); j++) {  
    System.out.println(blogPostBO.getPosts().get(j).getContent());  
}  
  
System.out.println("=====这是对象分割线=====");  
}  
}
```

测试结果

```
2018-10-17 20:49:22,444 [main] DEBUG [com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListByBlog] - ==> Preparing: SELECT b.id, b.title, b.author_id, p.id as post_id, p.content FROM blog b LEFT JOIN post p ON b.id = p.blog_id  
2018-10-17 20:49:22,514 [main] DEBUG [com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListByBlog] - ==> Parameters:  
2018-10-17 20:49:22,554 [main] TRACE [com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListByBlog] - <== Columns: id, title, author_id, post_id, post_blog_id, post_content  
2018-10-17 20:49:22,554 [main] TRACE [com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListByBlog] - <== Row: 1, 小明的博客, 1, 1, 0, <<BLOB>>  
2018-10-17 20:49:22,561 [main] TRACE [com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListByBlog] - <== Row: 1, 小明的博客, 1, 2, 1, 0, <<BLOB>>  
2018-10-17 20:49:22,561 [main] TRACE [com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListByBlog] - <== Row: 1, 小明的博客, 1, 3, 1, 0, <<BLOB>>  
2018-10-17 20:49:22,561 [main] TRACE [com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListByBlog] - <== Row: 2, 小王的博客, 2, 4, 2, 0, <<BLOB>>  
2018-10-17 20:49:22,561 [main] TRACE [com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListByBlog] - <== Row: 2, 小王的博客, 2, 5, 2, 0, <<BLOB>>  
2018-10-17 20:49:22,561 [main] TRACE [com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListByBlog] - <== Row: 3, 大牛的博客, 3, null, null, null, <<BLOB>>  
2018-10-17 20:49:22,569 [main] TRACE [com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListByBlog] - <== Row: 4, 黑子的博客, 4, null, null, null, <<BLOB>>  
2018-10-17 20:49:22,569 [main] DEBUG [com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListByBlog] - <== Total: 7  
2018-10-17 20:49:22,577 [main] DEBUG [org.apache.ibatis.transaction.jdbc.JdbcTransaction] - Resetting autocommit to true on JDBC Connection [com.mysql.jdbc.JDBC4Connection@1a20270e]  
2018-10-17 20:49:22,577 [main] DEBUG [org.apache.ibatis.transaction.jdbc.JdbcTransaction] - Closing JDBC Connection [com.mysql.jdbc.JDBC4Connection@1a20270e]  
2018-10-17 20:49:22,577 [main] DEBUG [org.apache.ibatis.datasource.pooled.PooledDataSource] - Returned connection 438314766 to pool.  
小明的博客  
小明-第1篇测试啦啦啦  
小明-第2篇测试啦啦啦  
第3篇测试啦啦啦  
=====这是对象分割线=====  
小王的博客  
小王-我是大王，大家好  
大牛-我是一只牛  
=====这是对象分割线=====  
大牛的博客  
=====这是对象分割线=====  
黑子的博客  
=====这是对象分割线=====
```

https://blog.csdn.net/weixin_37139197

可以看出，已经获取到了。

2.1.5 结果合并原理

刚开始的时候，我们获取到的结果在数据库中应该是这样子的

2. 运行时动态库: not found 及介绍-linux的-Wl,-rpath命令(3508)

3. JDBC详解系列（一）之流程(3348)

4. Paho - MQTT C Client的实现(3036)

5. MyBatis动态SQL（认真看看，以后写SQL就爽多了）(2198)

评论排行榜

1. MyBatis动态SQL（认真看看，以后写SQL就爽多了）(14)

2. mybatis 多个接口参数的注解使用方式(@Param)(7)

3. mybatis源码-解析配置文件（二）之解析的流程(3)

4. mybatis源码-原来resultMap解析完是这样(3)

5. Java设计模式-建造者(Builder)模式(2)

推荐排行榜

1. MyBatis动态SQL（认真看看，以后写SQL就爽多了）(15)

2. HashMap是如何工作的(5)

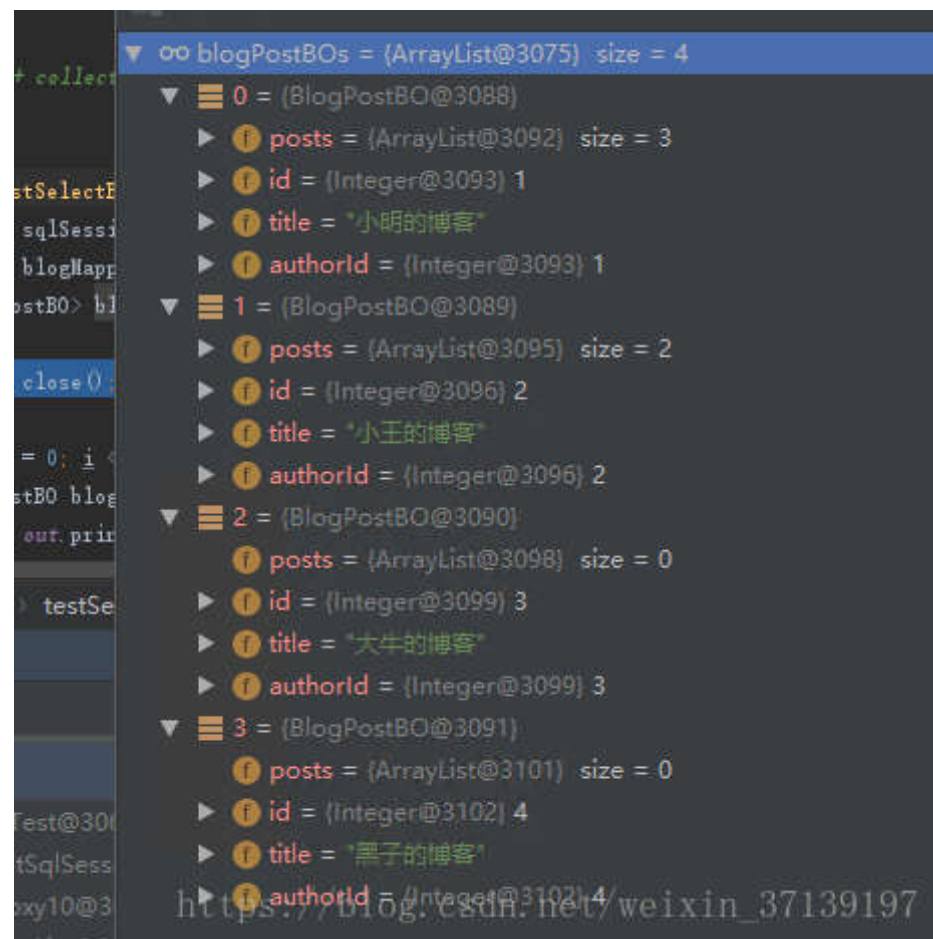
3. 扒一扒：Java 中的枚举(4)

4. mybatis源码-原来resultMap解析完是这样(4)

5. mybatis源码-Mapper解析之SQL 语句节点解析（一条语句对应一个MappedStatement)(2)

<input type="checkbox"/>	id	title	author_id	post_id	post_blog_id	post_draft	post_content	
<input type="checkbox"/>	1	小明的博客	1	1	1	0	小明-第1篇测试啦啦啦	29B
<input type="checkbox"/>	1	小明的博客	1	2	1	0	小明-第2篇测试啦啦啦	29B
<input type="checkbox"/>	1	小明的博客	1	3	1	0	第3篇测试啦啦啦	22B
<input type="checkbox"/>	2	小王的博客	2	4	2	0	小王-我是大王， 大家好	32B
<input type="checkbox"/>	2	小王的博客	2	5	2	0	大牛-我是一只牛	22B
<input type="checkbox"/>	3	大牛的博客	3	(NULL)	(NULL)	(NULL)	(NULL)	0K
<input type="checkbox"/>	4	黑子的博客	4	(NULL)	(NULL)	(NULL)	(NULL)	0K

那么，结果是怎么变成下面这样的呢？



2.1.5.1 合并的依据

mybatis 在处理结果时，会判断对象是否相同，如果相同则会将结果与之前的结果进行合并。

那么，结果是否相同是如何判断的呢？

首先是通过 **id** 的比较，如果 **id** 没有配置则比较每一个字段（此时，只要有一个字段不同的对象不同）。

2.1.5.2 id 的作用

在生成的 **XML** 中，如果有主键，一般会生成一个对应的 **id** 属性。

而 **id** 属性就可以用来判断获取到的数据是否属于同一个对象。

在以上的例子中，数据库中查询出来有三个 **id=1**，则 **mybatis** 在处理结果时就可以知道这三条数据对应相同的对象，从而将他们合并。

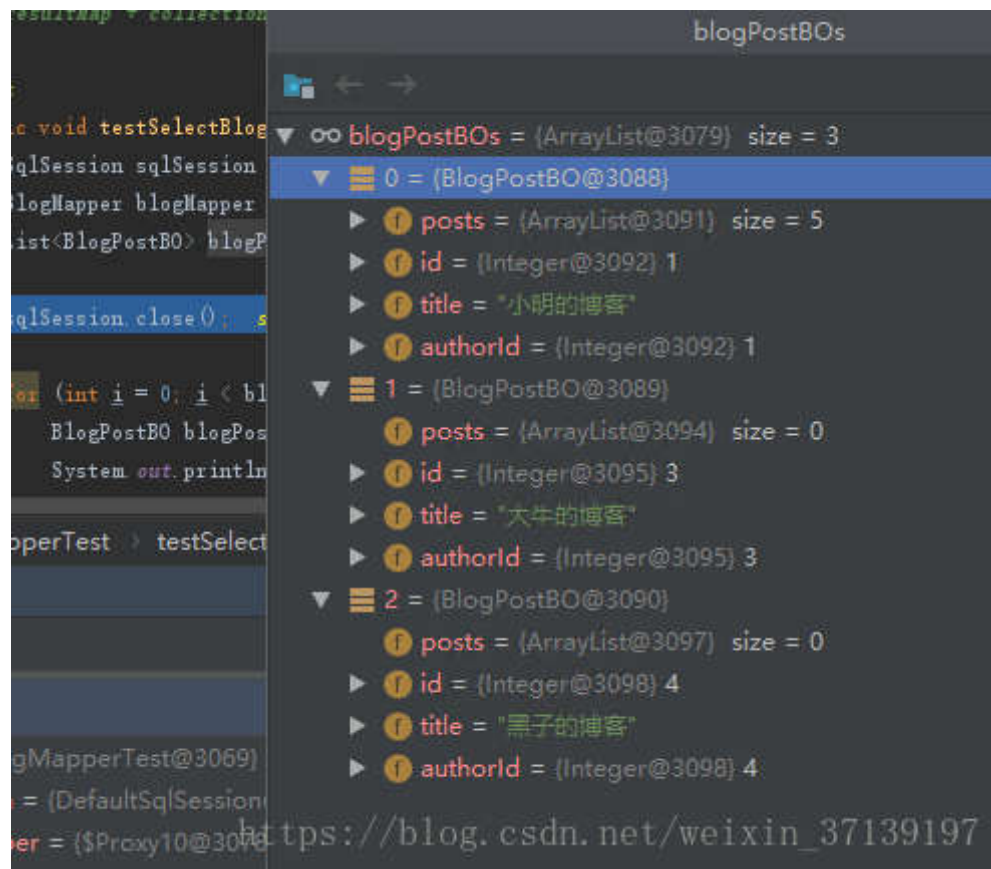
2.1.5.3 id 的作用验证

我们可以测试一下，将 **BlogMapper.xml** 的 **BaseResultMap** 改成 **title** 为 **id**，然后将数据库改一下

<input type="checkbox"/>	id	title	auth...
<input type="checkbox"/>	1	小明的博客	1
<input type="checkbox"/>	2	小明的博客	2
<input type="checkbox"/>	3	大牛的博客	3
<input type="checkbox"/>	4	黑子的博客	4
*	(Auto)	(NULL)	(NULL)

那么，此时博客 **id=1** 和 博客 **id=2** 的 **title** 都为“小明的博客”，那么，按照以上的推论，此两博客的对象应该合二为一。

真相揭晓



小王的博客消失了，小明的博客发表文章数量从 3 变为 5，因为合并了小王博客发布的文章数量。

2.1.5.4 建议

建议尽量配置 id 的属性，如果没有这个属性，则 mybatis 在进行结果合并时效率会低很多。

2.2 collection 嵌套查询方式

2.2.1 创建结果实体类

我们需要创建一个 `BlogPostBO`，

```
public class BlogPostBO extends Blog {

    private List<Post> posts;
```



```
public List<Post> getPosts() {  
    return posts;  
}  
  
public void setPosts(List<Post> posts) {  
    this.posts = posts;  
}  
}
```

该类继承于 `Blog` 类，并多了一个链表成员变量 `posts`，我们后续获取到的发布的文章都在此链表中。

2.2.2 创建结果集

结果集中多了一个 **select** 的属性，同时，需要相应的指定 **posts** 和 **column**

```
<resultMap id="BlogPostCustom" type="com.homejim.mybatis.entity.BlogPostBO"  
extends="BaseResultMap">  
    <collection property="posts" column="{blogId=id}"  
        select="com.homejim.mybatis.mapper.PostMapper.selectPostByBlogId" fetchType="lazy" />  
</resultMap>
```

重要说明：

1. **column** 中，需要填写 `{属性名=列名(别名)}` 的方式（如果只有一个参数，也可以直接填列名传入参数的即可）。如果是传递多个参数，则是 `{属性名1=列名1, 属性名2=列名2}`。
2. **select** 属性值 `com.homejim.mybatis.mapper.PostMapper.selectPostByBlogId`，对应的 **parameterType** 是 `java.util.Map`（也可以不填）。
3. **fetchType** 的属性值 **lazy**(延迟加载，还需要相应的配置) 或 **eager**。（更详细的参考我上篇文章[\[mybatis-高级结果映射之一对一\]](https://www.cnblogs.com/homejim/p/9808847.html)）

2.2.3 创建对应的方法和XML

主方法

```
/**  
 * 获取博客及其发布的内容 一对多:延迟加载  
 * @return
```

```
*/  
List<BlogPostBO> selectBlogAndPostListLazy();
```

主方法对应的 XML

```
<select id="selectBlogAndPostListLazy" resultMap="BlogPostCustom">  
    SELECT  
        b.id,  
        b.title,  
        b.author_id  
    FROM blog b  
</select>
```

嵌套方法

```
/**  
 * 根据博客 id 获取发布的文章信息  
 * @param blogId 博客 id  
 * @return  
 */  
List<Post> selectPostByBlogId(int blogId);
```

嵌套方法对应 XML

```
<select id="selectPostByBlogId" parameterType="java.util.Map" resultMap="ResultMapWithBLOBs">  
    select  
    <include refid="Base_Column_List" />  
    ,  
    <include refid="Blob_Column_List" />  
    from post  
    where blog_id = #{blogId,jdbcType=INTEGER}  
</select>
```

2.2.4 测试

```
/**  
 * resultMap + collection 一对多映射，嵌套查询  
 */  
@Test  
public void testSelectBlogAndPostListByBlogLazy() {
```

```

SqlSession sqlSession = sqlSessionFactory.openSession();
BlogMapper blogMapper = sqlSession.getMapper(BlogMapper.class);
List<BlogPostBO> blogPostBOs = blogMapper.selectBlogAndPostListLazy();

for (int i = 0; i < blogPostBOs.size(); i++) {
    BlogPostBO blogPostBO = blogPostBOs.get(i);
    System.out.println(blogPostBO.getTitle());
    List<Post> posts = blogPostBO.getPosts();

    for (int j = 0; j < posts.size(); j++) {
        System.out.println(blogPostBO.getPosts().get(j).getContent());
    }

    System.out.println("=====这是对象分割线=====");
}
}

```

```

Thu Oct 18 09:43:24 CST 2018 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5
[org.apache.ibatis.datasource.pooled.PooledDataSource] - Created connection 89448984.
[org.apache.ibatis.transaction.jdbc.JdbcTransaction] - Setting autocommit to false on JDBC Connection [com.mysql.jdbc.JDBC4Connection@554e218]
[com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListLazy] - ==> Preparing: SELECT b.id, b.title, b.author_id FROM blog b
[com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListLazy] - ==> Parameters:
[com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListLazy] - <= Columns: id, title, author_id
[com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListLazy] - <= Row: 1, 小明的博客, 1
[com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListLazy] - <= Row: 2, 小王的博客, 2
[com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListLazy] - <= Row: 3, 大牛的博客, 3
[com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListLazy] - <= Row: 4, 黑子的博客, 4
[com.homejim.mybatis.mapper.BlogMapper.selectBlogAndPostListLazy] - <= Total: 4
小明的博客
[com.homejim.mybatis.mapper.PostMapper.selectPostByBlogId] - ==> Preparing: select id, blog_id, draft, content from post where blog_id = ?
[com.homejim.mybatis.mapper.PostMapper.selectPostByBlogId] - ==> Parameters: 1(Long)
[com.homejim.mybatis.mapper.PostMapper.selectPostByBlogId] - <= Columns: id, blog_id, draft, content
[com.homejim.mybatis.mapper.PostMapper.selectPostByBlogId] - <= Row: 1, 1, 0, <<BLOB>>
[com.homejim.mybatis.mapper.PostMapper.selectPostByBlogId] - <= Row: 2, 1, 0, <<BLOB>>
[com.homejim.mybatis.mapper.PostMapper.selectPostByBlogId] - <= Row: 3, 1, 0, <<BLOB>>
[com.homejim.mybatis.mapper.PostMapper.selectPostByBlogId] - <= Total: 3
小明-第1篇测试啦啦啦啦
小明-第2篇测试啦啦啦啦
第3篇测试啦啦啦啦
=====这是对象分割线=====
小王的博客
[com.homejim.mybatis.mapper.PostMapper.selectPostByBlogId] - ==> Preparing: select id, blog_id, draft, content from post where blog_id = ?
[com.homejim.mybatis.mapper.PostMapper.selectPostByBlogId] - ==> Parameters: 2(Long)
[com.homejim.mybatis.mapper.PostMapper.selectPostByBlogId] - <= Columns: id, blog_id, draft, content

```

可以看到，只有在需要使用对象的时候，才会进行延迟加载。

3 代码

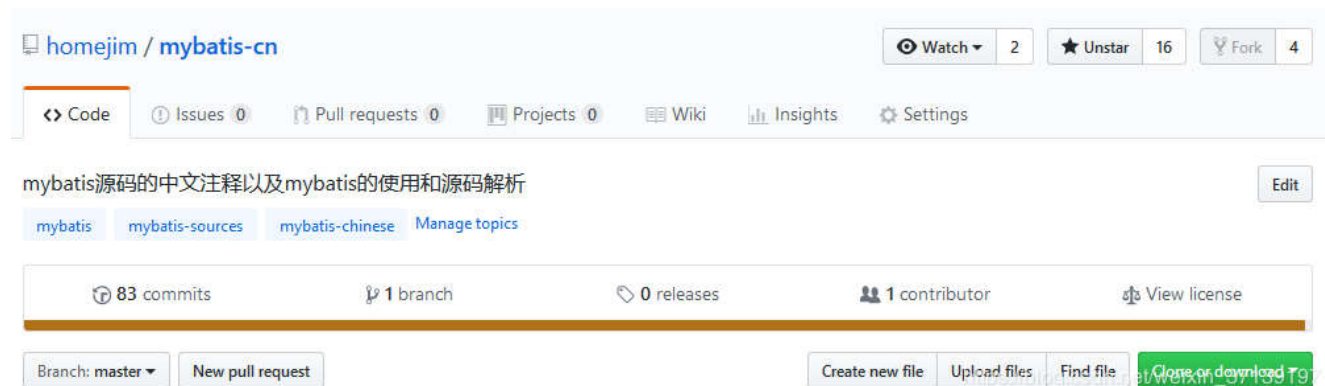
我的 [Github:mybatis-mapping](#)

一起学 mybatis

你不想来学习 mybatis? 学习其使用和源码呢? 那么，在博客园关注我吧！！

我自己打算把这个源码系列更新完毕，同时会更新相应的注释。快去 star 吧！！

[mybatis最新源码和注释](#)



作者：[阿进的写字台](#)

出处：<https://www.cnblogs.com/homejim/>

本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，否则保留追究法律责任的权利。

分类：[mybatis-使用](#)

标签：[Java](#), [mybatis](#), [mybatis使用](#)

好文要顶

关注我

收藏该文



[阿进的写字台](#)[关注 - 4](#)[粉丝 - 47](#)[+加关注](#)« 上一篇: [mybatis-高级结果映射之一对一](#)» 下一篇: [Kafka 入门三问](#)

posted @ 2018-10-18 10:41 阿进的写字台 阅读(292) 评论(2) 编辑 收藏

#1楼 2018-10-18 10:50 Java使徒



获益颇多，希望楼主能够坚持更新

支持(1) 反对(0)



#2楼[楼主] 2018-10-18 10:52 阿进的写字台



@ Java使徒

打算写一些列的使用，还有一系列的源码解析

支持(1) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)**注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。****【推荐】**超50万VC++源码: 大型组态工控、电力仿真CAD与GIS源码库!**【活动】**华为云12.12会员节 云产品1折起 满额送Mate20 点击抢购**【推荐】**服务器100%基准CPU性能，1核1G首年168元，限时特惠!



相关博文：

- 【Mybatis高级映射】一对一映射、一对多映射、多对多映射
- mybatis关系映射之一对多和多对一
- 【MyBatis学习09】高级映射之一对多查询
- mybatis一对一、一对多、多对多映射
- mybatis 高级映射 - 一对多查询 - collection

最新新闻：

- 王兴：20年到 C，20年到 B
 - 硅谷是个什么谷（第二十五章）：更好的世界
 - 戴威的董事会股东的签字权 谁搞垮了ofo的资金和前途？
 - 携程有忧，建章难退
 - 史玉柱重出江湖三年：属于他的时代，远去了
- » 更多新闻...

Copyright ©2018 阿进的写字台