Atitit mybatis 返回多个数据集 类似json结构

目录

[1.1. Union只能结构相同 1](#_Toc21092)

[1.2. resultMap 1](#_Toc17441)

[1.3. 关联 有一个"类型的关系 2](#_Toc14601)

[1.4. 多对多关联 3](#_Toc17641)

## Union只能结构相同

## resultMap

* constructor - 类在实例化时,用来注入结果到构造方法中
  + idArg - ID 参数;标记结果作为 ID 可以帮助提高整体效能
  + arg - 注入到构造方法的一个普通结果
* id – 一个 ID 结果;标记结果作为 ID 可以帮助提高整体效能
* result – 注入到字段或 JavaBean 属性的普通结果
* association – 一个复杂的类型关联;许多结果将包成这种类型
  + 嵌入结果映射 – 结果映射自身的关联,或者参考一个
* collection – 复杂类型的集
  + 嵌入结果映射 – 结果映射自身的集,或者参考一个

## **关联** 有一个"类型的关系

<association property="author" column="blog\_author\_id" javaType="Author">

<id property="id" column="author\_id"/>

<result property="username" column="author\_username"/></association>

关联元素处理"有一个"类型的关系。比如,在我们的示例中,一个博客有一个用户。 关联映射就工作于这种结果之上。你指定了目标属性,来获取值的列,属性的 java 类型(很 多情况下 MyBatis 可以自己算出来) ,如果需要的话还有 jdbc 类型,如果你想覆盖或获取的 结果值还需要类型控制器。

关联中不同的是你需要告诉 MyBatis 如何加载关联。MyBatis 在这方面会有两种不同的 方式:

* 嵌套查询:通过执行另外一个 SQL 映射语句来返回预期的复杂类型。
* 嵌套结果:使用嵌套结果映射来处理重复的联合结果的子集。首先,然让我们来查看这个元素的属性。所有的你都会看到,它和普通的只由 select 和

<select id="selectBlog" resultMap="blogResult">

SELECT \* FROM BLOG WHERE ID = #{id}

</select>

<resultMap id="blogResult" type="Blog">

<association property="author" column="author\_id" javaType="Author" select="selectAuthor"/>

</resultMap>

<select id="selectAuthor" resultType="Author">

SELECT \* FROM AUTHOR WHERE ID = #{id}

</select>

## 多对多关联

面你已经看到了如何处理"有一个"类型关联。但是"有很多个"是怎样的?下面这 个部分就是来讨论这个主题的。

#### **集合**

<collection property="posts" ofType="domain.blog.Post">

<id property="id" column="post\_id"/>

<result property="subject" column="post\_subject"/>

<result property="body" column="post\_body"/></collection>

集合元素的作用几乎和关联是相同的。实际上,它们也很相似,文档的异同是多余的。 所以我们更多关注于它们的不同。

我们来继续上面的示例,一个博客只有一个作者。但是博客有很多文章。在博客类中, 这可以由下面这样的写法来表示:

private List posts;

集合的嵌套查询

首先,让我们看看使用嵌套查询来为博客加载文章。

<resultMap id="blogResult" type="Blog">

<collection property="posts" javaType="ArrayList" column="id" ofType="Post" select="selectPostsForBlog"/>

</resultMap>

<select id="selectBlog" resultMap="blogResult">

SELECT \* FROM BLOG WHERE ID = #{id}

</select>

<select id="selectPostsForBlog" resultType="Blog">

SELECT \* FROM POST WHERE BLOG\_ID = #{id}

</select>

这里你应该注意很多东西,但大部分代码和上面的关联元素是非常相似的。首先,你应 该注意我们使用的是集合元素。然后要注意那个新的"ofType"属性。这个属性用来区分 JavaBean(或字段)属性类型和集合包含的类型来说是很重要的。所以你可以读出下面这个 映射:

<collection property="posts" javaType