


😊 17、resultMap子元素

📖 微信公众号java-mindmap

id & result

例子 


```
<id property="id" column="post_id"/>
<result property="subject" column="post_subject"/>
```

结果映射最基本内容

都映射一个单独列的值到简单数据类型(字符 串,整型,双精度浮点数,日期等)的单独属性或字段

这两者之间的唯一不同是 id 表示的结果将是当比较对象实例时用到的标识属性

构造方法constructor

例子 

```
<constructor>
<idArg column="id" javaType="int"/>
<arg column="username" javaType="String"/>
<arg column="age" javaType="_int"/>
</constructor>
```

```
public class User {
    //...
    public User(Integer id, String username, int age) {
        //...
    }
    //...
}
```

对应构造方法 

```
<association property="author" column="blog_author_id"
javaType="Author">
<id property="id" column="author_id"/>
<result property="username" column="author_username"/>
</association>
```

例子 

```
public class User {
    //...
    private Author author;
    //...
}
```

对应类型关系 

一个博客有一个用户

关联association

📖 <http://www.mybatis.org>

关联的嵌套查询 

```
<resultMap id="blogResult" type="Blog">
<association property="author" column="author_id" javaType="Author"
select="selectAuthor"/>
</resultMap>
```

```
<select id="selectBlog" resultMap="blogResult">
SELECT * FROM BLOG WHERE ID = #{id}
</select>
```


```
<select id="selectAuthor" resultType="Author">
SELECT * FROM AUTHOR WHERE ID = #{id}
</select>
```

两个查询语句:一个来加载博客,另外一个来加载作者

```
<resultMap id="blogResult" type="Blog">
<id property="id" column="blog_id" />
<result property="title" column="blog_title"/>
<association property="author" column="blog_author_id" javaType="Author"
resultMap="authorResult"/>
</resultMap>
```

```
<resultMap id="authorResult" type="Author">
<id property="id" column="author_id"/>
<result property="username" column="author_username"/>
<result property="password" column="author_password"/>
<result property="email" column="author_email"/>
<result property="bio" column="author_bio"/>
</resultMap>
```

关联的嵌套结果 

注意  id元素在嵌套结果映射中扮演着非常重要的角色。你应该总是指定一个或多个可以唯一标识结果的属性。

图片截图资料 

有关关联的部分建议通过给出的链接来学习，
应该有代码结合才能更好理解关联association的用法