

Mybatis 关联映射

1.配置 User 和 Role 的 1 对 1 映射关联

1.1: 改造实体类

在 User 类中添加 Role 属性: `private Role role;`

```
private Date modifyDate; //更新时间
private Role role;

public Role getRole() {
    return role;
}

public void setRole(Role role) {
    this.role = role;
}

public Integer getId() {
    return id;
}
```

1.2: 改造 UserMapper.xml

在 `UserMapper.xml` 中使用 `association` 配置一对一关联映射

```
<!-- 负责进行实体属性和表中字段映射 -->
<resultMap type="User" id="userList">
    <id property="id" column="id"/>
    <result property="userName" column="userName"/>
    <result property="address" column="address"/>
    <result property="userRoleName" column="roleName"/>
    <association property="role" column="userRole" javaType="Role">
        <id property="id" column="r_id"/>
        <result property="roleCode" column="r_roleCode"/>
        <result property="roleName" column="r_roleName"/>
    </association>
</resultMap>
```

```

<!-- 查询用户列表信息 不设置别名, 直接填写User 会报错 找不到User类型 -->
<select id="getUserList" resultMap="userList">
    SELECT U.*,R.id AS r_id,R.`roleCode` AS r_roleCode,R.`roleName` AS r_roleName
    FROM smbms_user AS U
    INNER JOIN smbms_role AS R ON U.`userRole`=R.`id`;
<!-- SELECT U.*,roleName FROM smbms_user AS U INNER JOIN smbms_role AS R ON U.`
</select>

```

至此 1 对 1 关联映射配置成功可用！

1.3: 改进

改进 1: 使用前缀

```

<resultMap type="User" id="userList">
    <id property="id" column="id"/>
    <result property="userName" column="userName"/>
    <result property="address" column="address"/>
    <result property="userRoleName" column="roleName"/>
    <association property="role" column="userRole" javaType="Role" columnPrefix="r_">
        <id property="id" column="id"/>
        <result property="roleCode" column="roleCode"/>
        <result property="roleName" column="roleName"/>
    </association>
</resultMap>

```

改进 2: 引用外部的 resultMap

```

mybatis-config.xml  User.java  UserMapper.xml  TestUser.java  *RoleMapper.xml
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 <mapper namespace="com.kawa.dao.RoleMapper">
6     <resultMap type="Role" id="roleMap">
7         <id property="id" column="id"/>
8         <result property="roleCode" column="roleCode"/>
9         <result property="roleName" column="roleName"/>
10        <result property="createdBy" column="createdBy"/>
11        <result property="creationDate" column="creationDate"/>
12        <result property="modifyBy" column="modifyBy"/>
13        <result property="modifyDate" column="modifyDate"/>
14    </resultMap>
15 </mapper>
16
15 = "roleName"/>
16 serRole" javaType="Role" columnPrefix="r_" resultMap="com.kawa.dao.RoleMapper.roleMap" </association>
17

```

改进 3：使用 extends

```
mapper namespace="com.kawa.dao.UserMapper">
  <!-- userMap -->
  <resultMap type="User" id="userMap">
    <id property="id" column="id"/>
    <result property="userName" column="userName"/>
    <result property="address" column="address"/>
    <result property="userRoleName" column="roleName"/>
  </resultMap>
  <!-- 负责进行实体属性和表中字段映射 -->
  <resultMap type="User" id="userList" extends="userMap">
    <association property="role" column="userRole" javaType="Role" columnPrefix="r_" resultMap="com.
  </resultMap>
```

2.配置 Role 和 User 的 1 对多关联映射

2.1：修改 Role 的实体类

添加 User 的 List 属性：private List<User> users = new ArrayList<User>();

```
public class Role {
    private Integer id;
    private String roleCode;
    private String roleName;
    private Integer createdBy;
    private String creationDate;
    private Integer modifyBy;
    private String modeifyDate;
    private List<User> users = new ArrayList<User>();

    public List<User> getUsers() {
        return users;
    }
    public void setUsers(List<User> users) {
        this.users = users;
    }

    public Integer getId() {
        return id;
    }
}
```

2.2：修改 RoleMapper.xml

在 RoleMapper.xml 中使用 collection 配置一对多关联映射

```

<resultMap type="Role" id="roleList" extends="roleMap">
  <collection property="users" column="id" columnPrefix="u_" ofType="User" resultMap="com.kawa.dao
</resultMap>

<!-- 查询所有角色列表信息 -->
<select id="getRoleList" resultMap="roleList">
  SELECT R.*,U.id AS u_id,U.`userName` AS u_userName,U.`address` AS u_address
  FROM smbms_user AS U INNER JOIN smbms_role AS R ON U.`userRole`=R.`id`
</select>

```

注意：一对一使用的是

javaType

一对多使用的是ofType