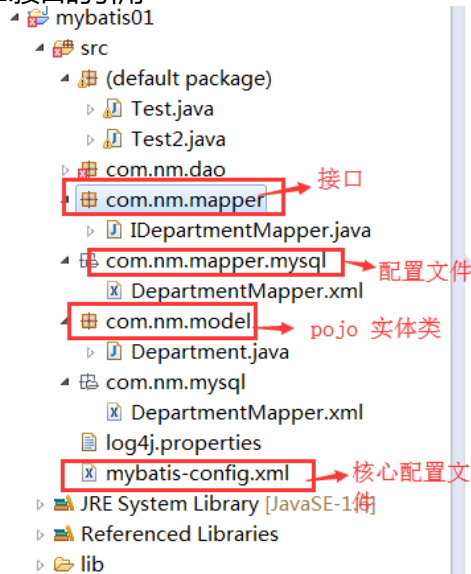


## mybatis02

笔记本： mybatis  
创建时间： 2019/6/26 20:30  
作者： qmys2018@163.com

更新时间： 2019/6/26 22:56

### 1.接口的引用



接口：

```
public interface IDepartmentMapper {  
    //删除  
    public int delete1(int id);  
}
```

mapper:

```
<mapper namespace="com.nm.mapper.IDepartmentMapper"> 是接口的全路径  
    <delete id="delete1" > delete1 和接口的方法名一致  
        delete from t_department where d_id = #{id5}  
    </delete>  
</mapper>
```

在mybatis-config.xml 中引用

```
<mapper resource="com/nm/mapper/mysql/DepartmentMapper.xml" />
```

测试：

```
IDepartmentMapper department=session.getMapper(IDepartmentMapper.class);  
int i=department.delete1(115);
```

### 2.配置不同返回值的方法 resultType

```
//删除  
public int delete1(int id);  
-----  
<delete id="delete1" > 省略  
    delete from t_department where d_id = #{id5}  
</delete>  
-----  
/*int i=department.delete1(115);  
    System.out.println(i);*/
```

```
//按照id来查询  
public Department findById(int id);  
-----
```

```
<select id="findById" resultType="com.nm.model.Department">
    select * from t_department where d_id=#{id6}
</select>
```

```
-----
//按照id查询
/*Department d=department.findById(3);
System.out.println(d.getDname());*/
```

```
//返回值是Map查询
public Map<String, Object> findById2(int id);
```

```
-----
<select id="findById2" resultType="map">
    select * from t_department where d_id=#{id6}
</select>
```

```
-----
//返回值是map
/*Map<String, Object> m=department.findById2(3);
System.out.println(m.get("dname")); //获取的值String 一定是表中字段的名 和类的属性名没有关系
System.out.println(m.get("remark"));*/
```

```
//返回值是list
public List<Department> findAll();
```

```
-----
<select id="findAll" resultType="com.nm.model.Department">
    select * from t_department
</select>
```

```
-----
/*//返回list
List<Department> list=department.findAll();
for(Department d:list){
    System.out.println(d.getDname());
}*/
```

```
//返回值是List<Map<String, Object>>
List<Map<String, Object>> findAll12();
```

```
-----
<select id="findAll12" resultType="map">
    select * from t_department
</select>
```

```
-----
//返回List<Map<String, Object>>
List<Map<String, Object>> l=department.findAll12();
for(Map<String, Object> m:l){
    System.out.println(m.get("dname")); //字段名
    System.out.println(m.get("remark"));
}
```

### 3. 参数的配置parameterType

```
public int insert1(Department d);

public int insert2(Map<String, Object> map);

public int insert3(String dname, String remark);

//注解
public int insert4(@Param("n") String dname, @Param("n2") String remark);
```

xml:

```
<!--#{dname} dname: 是department中属性名 不能更改 -->
<insert id="insert1" parameterType="com.nm.model.Department">
    insert into t_department values(d_id,#{dname},#{remark})
```

```

</insert>
<!-- a b 是map中put时key的值 map.put("a1", "yyy");-->
<insert id="insert2" >
insert into t_department values(d_id,#{a1},#{b})
</insert>
<!-- 高版本arg0 arg1 -->
<insert id="insert3" >
insert into t_department values(d_id,#{0},#{1})
</insert>
<!-- n n2 是接口中方法中 param后面的值@param("n") -->
<insert id="insert4" >
insert into t_department values(d_id,#{n},#{n2})
</insert>

```

#### 4.注解：

##### a.在接口中方法上加对应的注解

```

//删除
@Delete("delete from t_department where d_id=#{d_id}")
public int delete1(int id);
//按照id来查询
@Select("select * from t_department where d_id=#{d_id}")
public Department findById(int id);

```

##### b.一定要有mapper.xml

namespace 必填 里面的sql比用配置

#### 5.主键自动生成

```

<insert id="insert1" parameterType="com.nm.model.Department" useGeneratedKeys="true" keyProperty="d_id">
insert into t_department(dname,remark) values(#{dname},#{remark})
</insert>

```

#### 6.properties文件引用

##### src中创建jdbc.properties文件

```

driver=com.mysql.jdbc.Driver
url=jdbc:mysql://localhost:3306/xm
username=root
password=12345

```

##### 在config中引用

```

<!--引入properties文件 --> 放在最上面 注意标记的顺序 将光标放到configuration 上会有提示信息
<properties resource="jdbc.properties"></properties>

```

#### 7.别名

##### 1) 在mybatis-config中

```

<typeAliases>
<typeAlias type="com.nm.model.Department" alias="d"/> d就是我们自定义的别名
</typeAliases>
parameterType="d"

```

##### 2) 在mybatis-config中

```

<typeAliases>
<package name="com.nm.model"/> 包名
</typeAliases>
类的别名需要在类上面用注解实现配置
@Alias("dep")
public class Department {
xml中:
parameterType="dep"

```

##### 3) 在mybatis-config中

```

<typeAliases>
<package name="com.nm.model"/> 包名
</typeAliases>

```

这个包中所有的类的别名都是类名的小写形式