

# 深入parameterType（广泛应用于实际开发）

---

## 一、基本介绍

**parameterType** 属性用于定义对应的mapper接口方法接受的参数类型，取值可以是

- 1、基本类型，引用类型     :         String 类型，int
- 2、**复杂数据类型**:             实体类类型（POJO 类），Map

## 二、基本数据类型传参

```
<update id="updateUser" parameterType="java.lang.String">
    UPDATE tb_user SET user_name=#{userName},user_sex=#{userSex}
    WHERE user_id=#{userId}
</update>
```

## 三、pojo 包装对象

开发中通过 pojo 传递查询条件，查询条件是综合的查询条件，不仅包括用户查询条件还包括其它的查询条件（比如将用户购买商品信息也作为查询条件），这时可以使用包装对象传递输入参数。

需求：根据用户名查询用户信息，查询条件放到 QueryVo 的 user 属性中。

### 1、编写 QueryVo

```
public class QueryVo implements Serializable {
    private User user;
    public User getUser() {
        return user;
    }
    public void setUser(User user) {
        this.user = user;
    }
}
```

## 2、编写持久层

```
public interface IUserDao {  
    /**  
     * 根据 QueryVo 中的条件查询用户  
     * @param vo  
     * @return  
     */  
    List<User> findByVo(QueryVo vo);  
}
```

## 3、持久层接口的映射 文件

```
<!-- 根据queryVo的条件查询用户 -->  
<select id="findUserByVo" parameterType="com.itheima.domain.QueryVo"  
resultMap="com.itheima.domain.User">  
    select * from user where username like #{user.username}  
</select>
```

## 4、测试

```
@Test  
public void testFindByVo(){  
    QueryVo vo = new QueryVo();  
    User user = new User();  
    user.setUserName("%王%");  
    vo.setUser(user);  
    //5.执行查询一个方法  
    List<User> users = userDao.findUserByVo(vo);  
    for(User u : users){  
        System.out.println(u);  
    }  
}
```