



河北师范大学软件学院
Software College of Hebei Normal University



MyBatis

第七讲 MyBatis注解



Java与移动智能设备开发



- MyBatis注解方式就是将SQL语句直接写在映射接口上
- 优点：对于需求简单的系统，效率较高
- 缺点：当SQL发生变化时，需要重新编译代码
- 一般情况下不建议使用注解方式



- 使用注解方式时，在主配置文件中关联映射配置时应该使用如下方法

```
<mappers>  
    <mapper class="com.mybatis.mapper.UserMapper"/>  
</mappers>
```



1 @Select注解

2 @Insert注解

3 @Update注解

4 @Delete注解



- 在映射接口的查询方法上添加如下注解

```
@Select("select id, user_name userName, password"  
        + " from user where id = #{id}")  
public User selectById(Integer id);
```

- @Select注解的参数为字符串类型



■ 使用resultMap方式解决表字段名与属性名不一致问题

```
@Results({
    @Result(property="id", column="id", id=true),
    @Result(property="userName", column="user_name"),
    @Result(property="password", column="password")
})
@Select("select id, user_name, password"
        + " from user where id = #{id}")
public User selectById(Integer id);
```



- 注解@Results对应XML中的resultMap元素
- 注解@Result对应XML中的result元素，当参数中出现id=true时，就对应了id元素



- 从MyBatis3.3.1开始，映射接口中的多个方法可以共用@Results注解，首先要在@Results注解中加上id属性

```
@Results(id = "userResult", value= {  
    @Result(property="id",column="id",id=true),  
    @Result(property="userName", column="user_name"),  
    @Result(property="password", column="password")  
})  
@Select("select id, user_name, password"  
        + " from user where id = #{id}")  
public User selectById(Integer id);
```




- 在映射接口中添加查询方法，通过@ResultMap注解共用@Results注解
 - @ResultMap中参数的值与@Results注解中id属性的值一致

```
@ResultMap("userResult")
@Select("select * from user")
public List<User> selectAllUsers() ;
```



1 @Select注解

2 @Insert注解

3 @Update注解

4 @Delete注解



- 在映射接口的插入方法上添加如下注解

```
@Insert("insert into user(id, user_name, password)"  
        + "value("#{id}, #{userName}, #{password})")  
public int insert(User user);
```

- @Insert注解的参数为字符串类型



- 如果需要返回自增主键，需要增加@Options注解

```
@Insert("insert into user(id, user_name, password)"  
        + "value("#{id}, #{userName}, #{password})")  
@Options(useGeneratedKeys = true, keyProperty = "id")  
public int insert(User user);
```

- useGeneratedKeys和keyProperty属性同XML中相同



- 如果需要返回非自增主键，需要使用@SelectKey注解

```
@Insert("insert into user(id, user_name, password)"
      + "value("#{id}, #{userName}, #{password})")
@SelectKey(statement = "SELECT LAST_INSERT_ID()",
           keyProperty = "id",
           resultType = Integer.class,
           before = false)
public int insert(User user);
```



- @SelectKey注解与前面的selectKey元素基本一致，其中before为false时功能等同于order="AFTER"，before为true时功能等同于order="BEFORE"



- 1 @Select注解
- 2 @Insert注解
- 3 **@Update注解**
- 4 @Delete注解



- 在映射接口的更新方法上添加如下注解

```
@Update("update user set user_name=#{userName},"  
        + "password=#{password} "  
        + "where id=#{id}")  
public int updateById(User user);
```

- @Update注解的参数为字符串类型



1 @Select注解

2 @Insert注解

3 @Update注解

4 **@Delete注解**



- 在映射接口的删除方法上添加如下注解

```
@Delete("delete from user where id=#{id}")  
public int deleteById(Integer id);
```

- @Delete注解的参数为字符串类型



- MyBatis的两种映射配置的基本用法
 - XML方式、注解方式
- 了解四种基本的注解用法
 - @Select、@Insert、@Update、@Delete



THANK YOU
