链滴

MyBatis 通用 Mapper 研究 & 实现（一）

作者：lanjian

原文链接：https://ld246.com/article/1478505694685

来源网站：链滴

<h2>优点?</h2>
<a class="anchor" id="支持单表操作-两张表联查操作-不支持大于两张表操作_4" href="http:/
git.oschina.net/lan0217/mybatis-mapper#支持单表操作-两张表联查操作-不支持大于两张表操作_
">支持单表操作,两张表联查操作,不支持
于两张表操作
1.支持多种语句操作,如eq,ne,ge,gt,le,lt,is null,not null,is empty,not empty,in,not in,以上操作
为AND,目前不支持OR
2.支持两张表联查,依旧不写sql,同时也支持以上多种操作符。只需要简单配置一下注解即可。</p

3.支持字段别名,cas需求就很好处理了哦。对同一个字段可以取别名多次set。
4.有工具类,直接生成数据库建表文件,对象建模。
<h2><a class="anchor" id="缺点-_5" href="http://git.oschina.net/lan0217/mybatis-mapper#
点-_5">缺点?</h2>
目前id方案不支持数据库自增,采取编码set
d的方案,后续会实现id生成器。
没实现oracle。。。只实现了mysql数据
支持。
<h2><a class="anchor" id="使用方式_6" href="http://git.oschina.net/lan0217/mybatis-mappe
#使用方式_6">使用方式</h2>
参考如下几个实现类
1.QueryBuilder 实现查询业务
2.UpdateBuilder 实现更新业务
支持根据查询条件更新数据
3.InsertBuilder 实现插入业务
4.DeleteBuilder 实现删除业

<h2><a class="anchor" id="准备开始_7" href="http://git.oschina.net/lan0217/mybatis-mappe
#准备开始_7">准备开始</h2>
定义领域对象

<pre class="brush: java">@Alias("user_info")
@Table(table = "user_info")
public class UserInfo extends BasicModel<UserInfo> {
    @Column(length = "200", desc = "用户头像")
    private String poster;

@Column(length = "50")
private String nickname;

//省略get/set

}
</pre>

mybatis配置文件扫描领域对象,如想使用mybatis的扫描机制,切记数据库字段和对象字段要一致,
不一致请到xml里面去配置映射关系,目前没找到其他实现方式,下一版本直接在mybatis源码扩展功能,
量改动源码。

<pre class="brush: xml"><?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
        "http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>

```
    &lt;typeAliases&gt;
        &lt;package name="com.xxx.model"/&gt;
    &lt;/typeAliases&gt;
&lt;/configuration&gt;
```

定义一个mapper,不需要实现方法,泛型指定返回对象

```
public interface UserInfoMapper extends MysqlBaseMapper&lt;UserInfo&gt; {}
```

定义一个dao接口

```
public interface UserInfoDao extends MySqlBaseDao&lt;UserInfo&gt; {}
```

实现dao,对mapper进行绑定,指定mapper的class,dao也可以用传统的xml方式实现自己的业务,do绑定mapper为了调用mapper通用方法,也还原了大家经常使用mybatis的方式。

```
@Repository
public class UserInfoDaoImpl extends AbstractMySqlBaseDao&lt;UserInfo&gt; implements UserInfoDao {
    @Override
    public Class mapperClass() {
        return UserInfoMapper.class;
    }
}
```

spring配置里面加入如下,com.xxxx.mapper为项目定义mapper目录

```
&lt;bean class="org.mybatis.spring.mapper.MapperScannerConfiguer"&gt;
        &lt;property name="basePackage" value="com.xxxx.mapper"/&gt;
        &lt;property name="sqlSessionFactoryBeanName" value="sqlSessionFactory"/&gt;
 &lt;/bean&gt;

&lt;!-- myBatis Sql映射文件 --&gt;
&lt;bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean"&gt;
&lt;property name="dataSource" ref="multipleDataSource"/&gt;
&lt;property name="configLocation"&gt;
&lt;value&gt;classpath:mapper/entity-config.xml&lt;
value&gt;
```

</property>
<!--
依旧可以使用mybatis配置文件方式
<property name="mapperLocations" value="classpath*:mapper/*.xml"/>
-->
</bean> </pre>

<h2>开始使用</h2>
注意:所有property字符串都为java类字段义,非数据库字段。
查询单个对象

<pre class="brush: java">QueryBuilder queryBuilder = QueryBuilder.build(UserInfo.class).eq("d",1);
UserInfo userInfo = userInfoDao.one(queryBuilder);
</pre>

查询list

<pre class="brush: java">QueryBuilder queryBuilder = QueryBuilder.build(UserInfo.class).eq("tatus",1);
List<UserInfo> userInfos = userInfoDao.list(queryBuilder);
</pre>

查询总数

<pre class="brush: java">QueryBuilder queryBuilder = QueryBuilder.build(UserInfo.class).eq("tatus",1);
int count = userInfoDao.count(queryBuilder);
</pre>

查询当前页数据

<pre class="brush: java">QueryBuilder queryBuilder = QueryBuilder.build(UserInfo.class).pag(1).size(10).eq("status",1);
PageData page = userInfoDao.page(queryBuilder);
</pre>

更新数据

<pre class="brush: java">UpdateBuilder update = UpdateBuilder.build(UserInfo.class).set("las
_login_time",time).set("token", token).eq("id", 1);
int r = userInfoDao.update(update);
</pre>

插入数据

<pre class="brush: java">UserInfo userInfo = new UserInfo();
userInfo.setId(UUIDUtils.uuid());
userInfo.setToken(token);
userInfo.setChannel(channel);
userInfo.setStatus_at(StatusAt.active.getCode());
userInfo.setClient_version(clientVersion);
userInfo.setCredit(0);
userInfo.setIp(ip);
userInfo.setPhone(phone);
userInfo.setPassword(sourcePwd);
userInfo.setSalt(salt);
userInfo.setCreate_at(now());
userInfo.setUpdate_at(now());
userInfo.setOs(os);
InsertBuilder insert = InsertBuilder.build(UserInfo.class).set(userInfo);
int r = userInfoDao.insert(insert);
</pre>

删除数据

<pre class="brush: java">DeleteBuilder delete= DeleteBuilder.build(UserInfo.class).eq("id", 1);
int r = userInfoDao.delete(delete);</pre>