mybatis-dalgen 是 mybatis 代码自动生成，简化sql书写，避免大量动态sql。

**简介**

1. 解决问题 mybatis-maven-dalgen 后续简称 dalgen,解决mybatis代码自动生成的缺失.
2. 对比其他代码生成工具 \*mybatis-generator-maven-plugin    目前正在使用的代码生成工具,只能根据指定表结构反向生成do,mapper,mapper.xml且生成大量mybatis动态sql. \*dalgen    除能根据指定表结构生成 do,mapper,mapper.xml外,还能根据自定义sql 自动生成对应mapper接口中的方法等 ## 能做什么

**支持MySQL**

**支持分表**

\*具体参考 config.xml

**初始化生成的方法有**

\*insert,update,deleteByPrimary,getByPrimary四个简单地方法

**其他自定义SQL**

\*写在 operation中,执行 mvn mybatis:gen 后会自动帮您生成相关数据

**基本动态sql支持**

**自定义参数:**

\*指定 paramtype=primitive 会根据 sql中的 "#{...}" 生成原生的参数. \*指定 paramtype=object    参数为 DO 对象 \*其他参数类型不再支持,支持多了容易给维护带来困难

**定义返回结果:**

\*返回结果 可以指定 resultmap,resuttype \*resultmap 非变更类的 默认返回结果,不需要指定 \*reulttype 自定义返回类型,可以指定Map(需要写为全类) 原生类(String,Long等)大小写敏感 \*自定义resultmap, 有时候返回类型比较特殊,那么可以自己定义resultmap, dalgen 会帮您生成对应的类

**其他**

\*可以自由摸索

**暂不能做什么**

暂不支持OB,Oracle,后续版本会提供服务

**不会支持什么**

在 sql模板中不提供 mybatis特有的<sql>标签,避免复杂动态sql产生

**如何使用**

pom中添加依赖

[?](http://www.oschina.net/p/mybatis-dalgen/similar_projects?lang=19&sort=time&p=1)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <plugin>      <groupId>com.dalgen.mybatis-maven-plugin</groupId>      <artifactId>mybatis-maven-plugin</artifactId>      <version>1.3</version>      <configuration>  -- 可选 不填写 使用默认路径        <!--<templateDirectory>dalgen/templates</templateDirectory>-->          <outputDirectory>srcx</outputDirectory>          <!--<config>dalgen/config/config.xml</config>-->      </configuration></plugin> |

* 执行 mvn mybatis:gen 首先会初始化配置,减少了一步自己copy 配置文件的麻烦

[?](http://www.oschina.net/p/mybatis-dalgen/similar_projects?lang=19&sort=time&p=1)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 | config.xml 支持分页,支持OB,后续可能会考虑参数简写为 ?<?xml version="1.0" encoding="UTF-8"?><!-- ============================================================== --><!-- Master configuration file for auto-generation of iPaycore dal. --><!-- ============================================================== --><config>      <!-- ========================================================== -->      <!-- The typemap("Type Map") maps from one java type to another -->      <!-- java type. If you feel the original sql data type to java -->      <!-- type mapping is not satisfactory, you can use typemap to -->      <!-- convert it to a more appropriate one. -->      <!-- ========================================================== -->      <typemap from="java.sql.Date" to="java.util.Date"/>      <typemap from="java.sql.Time" to="java.util.Date"/>      <typemap from="java.sql.Timestamp" to="java.util.Date"/>      <typemap from="java.math.BigDecimal" to="Long"/>      <typemap from="byte" to="int"/>      <typemap from="short" to="int"/>        <!-- ========================================================== -->      <!-- datasource config  可以配置多个-->      <!-- ========================================================== -->      <database name="risk" class="org.gjt.mm.mysql.Driver" type="mysql">          <property name="url" value="jdbc:mysql://10.209.176.32:3406/xxpt\_rec\_dc"/>          <property name="userid" value="xxpt\_rec\_dc"/>          <property name="password" value="hello1234"/>          <property name="schema" value="xxpt\_rec\_dc"/>      </database>        <!--ob 配置 执行失败则需要 替换${java\_home}/jre/lib/security/ 下面的local\_policy.jar和US\_export\_policy.jar-->      <database name="fporgassetcenter" type="ob">          <property name="url" value="http://obconsole.test.alibaba-inc.com/ob-config/config.co?dataId=daily\_052"/>      </database>        <!-- ========project.name pom.xml中的值========================= -->      <!--<package value="com.oschina.${project.name}.common.dal.${database.name}.auto"/>-->      <package value="com.alibaba.recruit.datacenter.dal.${database.name}"/>        <!-- ========================================================== -->      <!-- 省略前置 支持多个 -->      <tablePrefix value="dc\_bg"/><!--长的放前面-->      <tablePrefix value="dc"/>      <tablePath value="${database.name}Tables/"/>      <!--分库分表规则  分表后缀 支持多个-->      <splitTableSuffix value="\_000"/></config> |

**table.xml 例子**

\*首次执行输出DC\_BG\_RISK\_SCAN 后DC\_BG\_RISK\_SCAN.xml会自动生成,您仅需要添加自己的sql即可

[?](http://www.oschina.net/p/mybatis-dalgen/similar_projects?lang=19&sort=time&p=1)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98 | DC\_BG\_RISK\_SCAN.xml<!DOCTYPE table SYSTEM "../table-config-1.1.dtd"><!-- sqlname逻辑表,用于生成对象   physicalName物理表,用于从数据中获取数据 --><table sqlname="DC\_BG\_RISK\_SCAN" physicalName="DC\_BG\_RISK\_SCAN">      <!--  特殊字符说明  &lt;&gt;   <> -->      <operation name="insert" paramtype="object" remark="插入表:DC\_BG\_RISK\_SCAN">          <selectKey resultType="java.lang.Long" keyProperty="id" order="AFTER">              SELECT              LAST\_INSERT\_ID()        </selectKey>          INSERT INTO DC\_BG\_RISK\_SCAN(              ID              ,NAME              ,RISK              ,DETAIL              ,EDU\_INFO              ,ID\_CARD\_NO              ,DETAIL\_URL              ,GMT\_CREATE              ,GMT\_MODIFIED          )VALUES(               #{id,jdbcType=BIGINT}              , #{name,jdbcType=VARCHAR}              , #{risk,jdbcType=CHAR}              , #{detail,jdbcType=VARCHAR}              , #{eduInfo,jdbcType=VARCHAR}              , #{idCardNo,jdbcType=VARCHAR}              , #{detailUrl,jdbcType=VARCHAR}              , #{gmtCreate,jdbcType=TIMESTAMP}              , #{gmtModified,jdbcType=TIMESTAMP}          )    </operation>        <operation name="update" paramtype="object" remark="更新表:DC\_BG\_RISK\_SCAN">          UPDATE DC\_BG\_RISK\_SCAN          SET              ID              = #{id,jdbcType=BIGINT}              ,NAME            = #{name,jdbcType=VARCHAR}              ,RISK            = #{risk,jdbcType=CHAR}              ,DETAIL          = #{detail,jdbcType=VARCHAR}              ,EDU\_INFO        = #{eduInfo,jdbcType=VARCHAR}              ,ID\_CARD\_NO      = #{idCardNo,jdbcType=VARCHAR}              ,DETAIL\_URL      = #{detailUrl,jdbcType=VARCHAR}              ,GMT\_CREATE      = #{gmtCreate,jdbcType=TIMESTAMP}              ,GMT\_MODIFIED    = #{gmtModified,jdbcType=TIMESTAMP}          WHERE              ID              = #{id,jdbcType=BIGINT}    </operation>        <operation name="deleteByPrimary" multiplicity="one" remark="根据主键删除数据:DC\_BG\_RISK\_SCAN">          DELETE FROM              DC\_BG\_RISK\_SCAN          WHERE              ID = #{id,jdbcType=BIGINT}    </operation>        <operation name="getByPrimary" multiplicity="one"  remark="根据主键获取数据:DC\_BG\_RISK\_SCAN">          SELECT \*          FROM DC\_BG\_RISK\_SCAN          WHERE              ID = #{id,jdbcType=BIGINT}    </operation>        <!--自定义resultMap-->      <resultmap name="myResultMap" type="MyResult">          <column name="name" javatype="String"/>          <column name="risk" javatype="String"/>      </resultmap>        <!-- idCardNoXX 自定义参数,需要指定类型,可以通过 jdbcType 也可以通过 javaType -->      <operation name="getMyResultMap" resultmap="myResultMap" remark="自定义ResultMap">          select              name,risk          from          DC\_BG\_RISK\_SCAN          where          id\_card\_no=#{idCardNo}          or          id\_card\_no=#{idCardNoXX,jdbcType=VARCHAR}          limit 1    </operation>        <!-- idCardNoXX 自定义参数,需要指定类型,可以通过 jdbcType 也可以通过 javaType -->      <operation name="getListParams" resultmap="myResultMap" remark="foreach支持">          select          name,risk          from          DC\_BG\_RISK\_SCAN          where          id\_card\_no=#{idCardNo}          and          name in        <foreach collection="names" item="name" index="index" open="(" close=")" separator=",">              #{name,jdbcType=VARCHAR}        </foreach>          limit 1    </operation>        <!-- idCardNoXX 自定义参数,需要指定类型,可以通过 jdbcType 也可以通过 javaType -->      <operation name="getListParamsMany" multiplicity="many" resultmap="myResultMap" remark="foreach支持 many">          select          name,risk          from          DC\_BG\_RISK\_SCAN          where          id\_card\_no=#{idCardNo}          and          name in        <foreach collection="names" item="name" index="index" open="(" close=")" separator=",">              #{name,jdbcType=VARCHAR}        </foreach>      </operation></table> |

**生成结果**

[?](http://www.oschina.net/p/mybatis-dalgen/similar_projects?lang=19&sort=time&p=1)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117 | RiskScanDOMapper.xml<?xml version="1.0" encoding="UTF-8" ?><!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd" ><mapper namespace="com.alibaba.recruit.datacenter.risk.dal.mapper.RiskScanDOMapper">      <!-- 自动生成,请修改 DC\_BG\_RISK\_SCAN.xml -->      <resultMap id="BaseResultMap"  type="com.alibaba.recruit.datacenter.risk.dal.dataobject.RiskScanDO">          <id column="ID" property="id" jdbcType="BIGINT" javaType="Long"/>          <result column="NAME" property="name" jdbcType="VARCHAR" javaType="String"/>          <result column="RISK" property="risk" jdbcType="CHAR" javaType="String"/>          <result column="DETAIL" property="detail" jdbcType="VARCHAR" javaType="String"/>          <result column="EDU\_INFO" property="eduInfo" jdbcType="VARCHAR" javaType="String"/>          <result column="ID\_CARD\_NO" property="idCardNo" jdbcType="VARCHAR" javaType="String"/>          <result column="DETAIL\_URL" property="detailUrl" jdbcType="VARCHAR" javaType="String"/>          <result column="GMT\_CREATE" property="gmtCreate" jdbcType="TIMESTAMP" javaType="java.util.Date"/>          <result column="GMT\_MODIFIED" property="gmtModified" jdbcType="TIMESTAMP" javaType="java.util.Date"/>      </resultMap>        <resultMap id="myResultMap"  type="com.alibaba.recruit.datacenter.risk.dal.resultmap.MyResult">          <result column="NAME" property="name" jdbcType="VARCHAR" javaType="String"/>          <result column="RISK" property="risk" jdbcType="CHAR" javaType="String"/>      </resultMap>        <sql id="Base\_Column\_List">          ID,NAME,RISK,DETAIL,EDU\_INFO,ID\_CARD\_NO,DETAIL\_URL,GMT\_CREATE,GMT\_MODIFIED    </sql>          <!--插入表:DC\_BG\_RISK\_SCAN-->      <insert id="insert" >          <selectKey resultType="java.lang.Long" keyProperty="id" order="AFTER">              SELECT              LAST\_INSERT\_ID()        </selectKey>          INSERT INTO DC\_BG\_RISK\_SCAN(              ID              ,NAME              ,RISK              ,DETAIL              ,EDU\_INFO              ,ID\_CARD\_NO              ,DETAIL\_URL              ,GMT\_CREATE              ,GMT\_MODIFIED          )VALUES(               #{id,jdbcType=BIGINT}              , #{name,jdbcType=VARCHAR}              , #{risk,jdbcType=CHAR}              , #{detail,jdbcType=VARCHAR}              , #{eduInfo,jdbcType=VARCHAR}              , #{idCardNo,jdbcType=VARCHAR}              , #{detailUrl,jdbcType=VARCHAR}              , #{gmtCreate,jdbcType=TIMESTAMP}              , #{gmtModified,jdbcType=TIMESTAMP}          )    </insert>        <!--更新表:DC\_BG\_RISK\_SCAN-->      <update id="update" >          UPDATE /\*MS-AUTODALGEN-DC-BG-RISK-SCAN-UPDATE\*/ DC\_BG\_RISK\_SCAN          SET              ID              = #{id,jdbcType=BIGINT}              ,NAME            = #{name,jdbcType=VARCHAR}              ,RISK            = #{risk,jdbcType=CHAR}              ,DETAIL          = #{detail,jdbcType=VARCHAR}              ,EDU\_INFO        = #{eduInfo,jdbcType=VARCHAR}              ,ID\_CARD\_NO      = #{idCardNo,jdbcType=VARCHAR}              ,DETAIL\_URL      = #{detailUrl,jdbcType=VARCHAR}              ,GMT\_CREATE      = #{gmtCreate,jdbcType=TIMESTAMP}              ,GMT\_MODIFIED    = #{gmtModified,jdbcType=TIMESTAMP}          WHERE              ID              = #{id,jdbcType=BIGINT}    </update>        <!--根据主键删除数据:DC\_BG\_RISK\_SCAN-->      <delete id="deleteByPrimary" >          DELETE /\*MS-AUTODALGEN-DC-BG-RISK-SCAN-DELETEBYPRIMARY\*/ FROM              DC\_BG\_RISK\_SCAN          WHERE              ID = #{id,jdbcType=BIGINT}    </delete>        <!--根据主键获取数据:DC\_BG\_RISK\_SCAN-->      <select id="getByPrimary" resultMap="BaseResultMap">          SELECT /\*MS-AUTODALGEN-DC-BG-RISK-SCAN-GETBYPRIMARY\*/  <include refid="Base\_Column\_List" />          FROM DC\_BG\_RISK\_SCAN          WHERE              ID = #{id,jdbcType=BIGINT}    </select>        <!--自定义ResultMap-->      <select id="getMyResultMap" resultMap="myResultMap">          select              name,risk          from          DC\_BG\_RISK\_SCAN          where          id\_card\_no=#{idCardNo}          or          id\_card\_no=#{idCardNoXX,jdbcType=VARCHAR}          limit 1    </select>        <!--foreach支持-->      <select id="getListParams" resultMap="myResultMap">          select          name,risk          from          DC\_BG\_RISK\_SCAN          where          id\_card\_no=#{idCardNo}          and          name in        <foreach collection="names" item="name" index="index" open="(" close=")" separator=",">              #{name,jdbcType=VARCHAR}        </foreach>          limit 1    </select>        <!--foreach支持 many-->      <select id="getListParamsMany" resultMap="myResultMap">          select          name,risk          from          DC\_BG\_RISK\_SCAN          where          id\_card\_no=#{idCardNo}          and          name in        <foreach collection="names" item="name" index="index" open="(" close=")" separator=",">              #{name,jdbcType=VARCHAR}        </foreach>      </select></mapper> |

[?](http://www.oschina.net/p/mybatis-dalgen/similar_projects?lang=19&sort=time&p=1)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57 | package com.alibaba.recruit.datacenter.risk.dal.mapper;import com.alibaba.recruit.datacenter.risk.dal.dataobject.RiskScanDO;import com.alibaba.recruit.datacenter.risk.dal.resultmap.MyResult;import java.util.List;/\*\*   \*The Table DC\_BG\_RISK\_SCAN.   \*风险扫描   \*/public interface RiskScanDOMapper{        /\*\*       \*desc:插入表:DC\_BG\_RISK\_SCAN.<br/>       \*descSql =  SELECT LAST\_INSERT\_ID() INSERT INTO DC\_BG\_RISK\_SCAN( ID ,NAME ,RISK ,DETAIL ,EDU\_INFO ,ID\_CARD\_NO ,DETAIL\_URL ,GMT\_CREATE ,GMT\_MODIFIED )VALUES( #{id,jdbcType=BIGINT} , #{name,jdbcType=VARCHAR} , #{risk,jdbcType=CHAR} , #{detail,jdbcType=VARCHAR} , #{eduInfo,jdbcType=VARCHAR} , #{idCardNo,jdbcType=VARCHAR} , #{detailUrl,jdbcType=VARCHAR} , #{gmtCreate,jdbcType=TIMESTAMP} , #{gmtModified,jdbcType=TIMESTAMP} )       \*@param RiskScanDO RiskScanDO       \*@return Long       \*/      Long insert(entity RiskScanDO);      /\*\*       \*desc:更新表:DC\_BG\_RISK\_SCAN.<br/>       \*descSql =  UPDATE DC\_BG\_RISK\_SCAN SET ID = #{id,jdbcType=BIGINT} ,NAME = #{name,jdbcType=VARCHAR} ,RISK = #{risk,jdbcType=CHAR} ,DETAIL = #{detail,jdbcType=VARCHAR} ,EDU\_INFO = #{eduInfo,jdbcType=VARCHAR} ,ID\_CARD\_NO = #{idCardNo,jdbcType=VARCHAR} ,DETAIL\_URL = #{detailUrl,jdbcType=VARCHAR} ,GMT\_CREATE = #{gmtCreate,jdbcType=TIMESTAMP} ,GMT\_MODIFIED = #{gmtModified,jdbcType=TIMESTAMP} WHERE ID = #{id,jdbcType=BIGINT}       \*@param RiskScanDO RiskScanDO       \*@return Long       \*/      Long update(entity RiskScanDO);      /\*\*       \*desc:根据主键删除数据:DC\_BG\_RISK\_SCAN.<br/>       \*descSql =  DELETE FROM DC\_BG\_RISK\_SCAN WHERE ID = #{id,jdbcType=BIGINT}       \*@param id id       \*@return Long       \*/      Long deleteByPrimary(Long id);      /\*\*       \*desc:根据主键获取数据:DC\_BG\_RISK\_SCAN.<br/>       \*descSql =  SELECT \*FROM DC\_BG\_RISK\_SCAN WHERE ID = #{id,jdbcType=BIGINT}       \*@param id id       \*@return RiskScanDO       \*/      RiskScanDO getByPrimary(Long id);      /\*\*       \*desc:自定义ResultMap.<br/>       \*descSql =  select name,risk from DC\_BG\_RISK\_SCAN where id\_card\_no=#{idCardNo} or id\_card\_no=#{idCardNoXX,jdbcType=VARCHAR} limit 1       \*@param idCardNo idCardNo       \*@param idCardNoXX idCardNoXX       \*@return MyResult       \*/      MyResult getMyResultMap(String idCardNo,String idCardNoXX);      /\*\*       \*desc:foreach支持.<br/>       \*descSql =  select name,risk from DC\_BG\_RISK\_SCAN where id\_card\_no=#{idCardNo} and name in #{name,jdbcType=VARCHAR} limit 1       \*@param idCardNo idCardNo       \*@param names names       \*@return MyResult       \*/      MyResult getListParams(String idCardNo,List<String> names);      /\*\*       \*desc:foreach支持 many.<br/>       \*descSql =  select name,risk from DC\_BG\_RISK\_SCAN where id\_card\_no=#{idCardNo} and name in #{name,jdbcType=VARCHAR}       \*@param idCardNo idCardNo       \*@param names names       \*@return List<MyResult>       \*/      List<MyResult> getListParamsMany(String idCardNo,List<String> names);} |

**增加分页支持**

[?](http://www.oschina.net/p/mybatis-dalgen/similar_projects?lang=19&sort=time&p=1)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68 | multiplicity="paging" --标记为此方法需要走分页查询  paging="QueryRisk"    --分页查询参数类名称                        --自动生成DAO类,通过Spring 自动扫描方法注入,不提供xml配置项目生成了.                        --DOMapper接口与DAO类区别在于对分页的支持    <operation name="getListParamsPage" multiplicity="paging" paging="QueryRisk" resultmap="myResultMap" remark="foreach支持 many">          select          name,risk          from          DC\_BG\_RISK\_SCAN          where          id\_card\_no=#{idCardNo}          and          name in        <foreach collection="names" item="name" index="index" open="(" close=")" separator=",">              #{name,jdbcType=VARCHAR}        </foreach>          order by gmt\_modified desc    </operation>Mapper.xml结果    <!--foreach支持 many pageCount-->      <select id="getListParamsPageCount"  resultType="int">          SELECT COUNT(\*) AS total FROM          DC\_BG\_RISK\_SCAN          where          id\_card\_no=#{idCardNo}          and          name in        <foreach collection="names" item="name" index="index" open="(" close=")" separator=",">              #{name,jdbcType=VARCHAR}        </foreach>        </select>      <!--foreach支持 many pageResult-->      <select id="getListParamsPageResult"  resultMap="myResultMap">          select          name,risk          from          DC\_BG\_RISK\_SCAN          where          id\_card\_no=#{idCardNo}          and          name in        <foreach collection="names" item="name" index="index" open="(" close=")" separator=",">              #{name,jdbcType=VARCHAR}        </foreach>          order by gmt\_modified desc          limit #{startRow},#{limit}    </select>DOMapper.java     /\*\*       \* desc:foreach支持 many.<br/>       \* descSql =  select name,risk from DC\_BG\_RISK\_SCAN where id\_card\_no=#{idCardNo} and name in #{name,jdbcType=VARCHAR} order by gmt\_modified desc       \* @param queryRisk queryRisk       \* @return int       \*/      int getListParamsPageCount(QueryRiskPage queryRisk);      /\*\*       \* desc:foreach支持 many.<br/>       \* descSql =  select name,risk from DC\_BG\_RISK\_SCAN where id\_card\_no=#{idCardNo} and name in #{name,jdbcType=VARCHAR} order by gmt\_modified desc       \* @param queryRisk queryRisk       \* @return List<MyResult>       \*/      List<MyResult> getListParamsPageResult(QueryRiskPage queryRisk);    DAO.java      /\*\*       \* desc:foreach支持 many.<br/>       \* descSql =  select name,risk from DC\_BG\_RISK\_SCAN where id\_card\_no=#{idCardNo} and name in #{name,jdbcType=VARCHAR} order by gmt\_modified desc       \* @param queryRisk queryRisk       \* @return QueryRiskPage       \*/      public QueryRiskPage getListParamsPage(QueryRiskPage queryRisk){          int total = riskScanDOMapper.getListParamsPageCount(queryRisk);          if(total>0){              queryRisk.setDatas(riskScanDOMapper.getListParamsPageResult(queryRisk));          }          queryRisk.setTotal(total);          return queryRisk;      }  如需分页会自动创建分页类 BasePage.java |