链滴

求解：PageHelper 分页不生效问题！

作者：ronger
原文链接：https://ld246.com/article/1527084692352
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

pom.xml

```
<parent>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-parent</artifactId>
 <version>1.5.13.RELEASE</version>
 <relativePath/> <!\-\- lookup parent from repository -->
</parent>

<properties>
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
 <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
 <java.version>1.8</java.version>
 <spring.boot.version>1.5.13.RELEASE</spring.boot.version>
</properties>

<dependencies>
  <!\-\- redis -->
  <dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-data-redis</artifactId>
 </dependency>  <!\-\- oracle驱动 -->
  <dependency>
 <groupId>com.oracle</groupId>
 <artifactId>ojdbc6</artifactId>
 <version>11.2.0.3</version>
 </dependency>  <!\-\- jdbc -->
  <dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-jdbc</artifactId>
 </dependency>  <!\-\- web -->
  <dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-web</artifactId>
 </dependency>  <!\-\- mybatis -->
  <dependency>
 <groupId>org.mybatis.spring.boot</groupId>
 <artifactId>mybatis-spring-boot-starter</artifactId>
 <version>1.3.2</version>
 </dependency>  <!\-\- tomcat依赖 -->
  <dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-tomcat</artifactId>
 <scope>provided</scope>
 </dependency>  <!\-\- jsp依赖 -->
  <dependency>
 <groupId>org.apache.tomcat.embed</groupId>
 <artifactId>tomcat-embed-jasper</artifactId>
 </dependency>  <!\-\- servlet-api -->
  <dependency>
 <groupId>javax.servlet</groupId>
 <artifactId>javax.servlet-api</artifactId>
 </dependency>  <!\-\- jstl -->
```

```
 <dependency>
<groupId>javax.servlet</groupId>
<artifactId>jstl</artifactId>
</dependency>  <!\-\- pagehelper -->
 <dependency>
<groupId>com.github.pagehelper</groupId>
<artifactId>pagehelper</artifactId>
<version>4.1.6</version>
</dependency>  <!\-\- pagehelper -->
<!--<dependency>--> <!--<groupId>org.springframework.boot</groupId>--> <!--<artifact
d>spring-boot-devtools</artifactId>--> <!--<scope>runtime</scope>--> <!--</dependenc
>--> <!\-\- lombok -->  <dependency>
<groupId>org.projectlombok</groupId>
<artifactId>lombok</artifactId>
<optional>true</optional>
</dependency>  <!\-\- mybatis-generator -->
 <dependency>
<groupId>org.mybatis.generator</groupId>
<artifactId>mybatis-generator-core</artifactId>
<version>1.3.5</version>
</dependency>  <!\-\- 阿里巴巴连接池Druid -->
 <dependency>
<groupId>com.alibaba</groupId>
<artifactId>druid</artifactId>
<version>1.1.9</version>
</dependency>  <!\-\- gson -->
 <dependency>
<groupId>com.google.code.gson</groupId>
<artifactId>gson</artifactId>
<version>2.8.4</version>
</dependency>  <!\-\- fastjson -->
 <dependency>
<groupId>com.alibaba</groupId>
<artifactId>fastjson</artifactId>
<version>1.2.47</version>
</dependency> <dependency> <groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>

<build>
 <!\-\- 程序编译时默认不会把 src/main/java 目录下文件添加到 jar 包中，需手动添加，
该配置影响 mybatis-generator 插件的使用,使用 mybatis-generator 时请注释 -->  <resources>
<resource> <directory>src/main/java</directory>
<includes> <include>**/*.xml</include>
</includes> </resource> </resources> <plugins> <plugin> <groupId>org.mybatis.generat
r</groupId>
<artifactId>mybatis-generator-maven-plugin</artifactId>
<version>1.3.5</version>
<configuration>  <!--配置文件的位置-->
 <configurationFile>src/main/resources/generatorConfig.xml</configurationFile>
 <!--允许移动生成的文件 -->
 <verbose>true</verbose>
```

```
  <!\-\- 是否覆盖 -->
  <overwrite>true</overwrite>
 </configuration> <dependencies> <dependency> <groupId>org.mybatis.generator</group
d>
 <artifactId>mybatis-generator-core</artifactId>
 <version>1.3.5</version>
 </dependency> </dependencies> </plugin> <plugin> <groupId>org.springframework.boo
</groupId>
 <artifactId>spring-boot-maven-plugin</artifactId>
 <configuration> <classifier>exec</classifier>
 </configuration> </plugin> </plugins></build>
```

mybatis-config.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>
 <properties> <property name="dialect" value="oracle" />
 </properties> <settings>  <!\-\- 这个配置使全局的映射器启用或禁用缓存。系统默认值是true，
置只是为了展示出来 -->
  <setting name="cacheEnabled" value="true" />
  <!\-\- 全局启用或禁用延迟加载。当禁用时，所有关联对象都会即时加载。 系统默认值是true，设
只是为了展示出来 -->
  <setting name="lazyLoadingEnabled" value="true" />
  <!\-\- 允许或不允许多种结果集从一个单独的语句中返回（需要适合的驱动）。 系统默认值是true
设置只是为了展示出来 -->
  <setting name="multipleResultSetsEnabled" value="true" />
  <!--使用列标签代替列名。不同的驱动在这方便表现不同。参考驱动文档或充分测试两种方法来决
所使用的驱动。 系统默认值是true，设置只是为了展示出来 -->
  <setting name="useColumnLabel" value="true" />
  <!--允许 JDBC 支持生成的键。需要适合的驱动。如果设置为 true 则这个设置强制生成的键被使用
尽管一些驱动拒绝兼容但仍然有效（比如
 Derby）。 系统默认值是false，设置只是为了展示出来 -->  <setting name="useGeneratedKeys"
value="false" />
  <!--配置默认的执行器。SIMPLE 执行器没有什么特别之处。REUSE 执行器重用预处理语句。BATCH
执行器重用语句和批量更新 系统默认值是SIMPLE，设置只是为了展示出来 -->
  <setting name="defaultExecutorType" value="SIMPLE" />
  <!--设置超时时间，它决定驱动等待一个数据库响应的时间。 系统默认值是null，设置只是为了展
出来 -->
  <setting name="defaultStatementTimeout" value="25000" />
 </settings> <plugins> <plugin interceptor="com.github.pagehelper.PageHelper">
 <property name="dialect" value="oracle" />
 <property name="offsetAsPageNum" value="true" />
 <property name="rowBoundsWithCount" value="true" />
 <property name="pageSizeZero" value="true" />
 <property name="reasonable" value="true" />
 </plugin> </plugins></configuration>
```

generatorConfig.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE generatorConfiguration
  PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"
 "http://mybatis.org/dtd/mybatis-generator-config\_1\_0.dtd">
<!\-\- 配置生成器 -->
<generatorConfiguration>
  <properties resource="mybatis-generator.properties" />

  <classPathEntry location="${driverClassName}" />

  <context id="oracleTables" defaultModelType="conditional" targetRuntime="MyBatis3" >

  <property name="autoDelimitKeywords" value="false"/>
  <!\-\- 生成的Java文件的编码 -->
  <property name="javaFileEncoding" value="UTF-8"/>
  <!\-\- -->
  <property name="suppressTypeWarnings" value="true" />
  <!\-\- 格式化java代码 -->
  <property name="javaFormatter" value="org.mybatis.generator.api.dom.DefaultJavaFormat
er"/>
  <!\-\- 格式化XML代码 -->
  <property name="xmlFormatter" value="org.mybatis.generator.api.dom.DefaultXmlFormatt
r"/>

  <property name="beginningDelimiter" value="`"/>
  <property name="endingDelimiter" value="`"/>

 <plugin type="org.mybatis.generator.plugins.SerializablePlugin" />
 <plugin type="org.mybatis.generator.plugins.CaseInsensitiveLikePlugin" />
 <plugin type="org.mybatis.generator.plugins.RowBoundsPlugin" />
 <plugin type="org.mybatis.generator.plugins.ToStringPlugin" />
 <plugin type="org.mybatis.generator.plugins.CachePlugin" />



  <plugin type="org.mybatis.generator.plugins.RenameExampleClassPlugin">
 <property name="searchString" value="Example$" />
 <property name="replaceString" value="Criteria" />
 </plugin>
  <!\-\- 注释 -->
  <commentGenerator>
 <property name="suppressAllComments" value="true" /><!\-\- 是否取消注释 -->
  <property name="suppressDate" value="true" /\> <!\-\- 是否生成注释代时间戳 -->
  </commentGenerator>

  <!\-\- 必须要有的，使用这个配置链接数据库
 @TODO:是否可以扩展
  -->
  <jdbcConnection driverClass="${jdbc.driverClass}" connectionURL="${jdbc.connectionURL}"
userId="${jdbc.userId}" password="${jdbc.password}">
  <!\-\- 这里面可以设置property属性，每一个property属性都设置到配置的Driver上 -->
  </jdbcConnection>

  <javaTypeResolver type="org.mybatis.generator.internal.types.JavaTypeResolverDefaultImpl
```

```
>
    <property name="forceBigDecimals" value="true"/>
  </javaTypeResolver>

  <javaModelGenerator targetPackage="${targetPackage}" targetProject="${targetProject}">
  <property name="constructorBased" value="false"/>


  <property name="enableSubPackages" value="true"/>

  <property name="immutable" value="false"/>

  <property name="trimStrings" value="true"/>
 </javaModelGenerator>

  <sqlMapGenerator targetPackage="${mapperPackage}" targetProject="${targetProject}">
  <property name="enableSubPackages" value="true"/>
 </sqlMapGenerator>

 <javaClientGenerator targetPackage="${daoPackage}" type="XMLMAPPER" targetProject="$
targetProject}">
  <property name="enableSubPackages" value="true"/>

  </javaClientGenerator>

   <table tableName="SYS_ROLE" domainObjectName="Role">

  <!\-\- 参考 javaModelGenerator 的 constructorBased属性-->
  <property name="constructorBased" value="false"/>

  <!\-\- 默认为false，如果设置为true，在生成的SQL中，table名字不会加上catalog或schema； --

  <property name="ignoreQualifiersAtRuntime" value="false"/>

  <!\-\- 参考 javaModelGenerator 的 immutable 属性 -->
  <property name="immutable" value="false"/>

  <!\-\- 指定是否只生成domain类，如果设置为true，只生成domain类，如果还配置了sqlMapGene
ator，那么在mapper XML文件中，只生成resultMap元素 -->
  <property name="modelOnly" value="false"/>

  <!\-\- 参考 javaModelGenerator 的 rootClass 属性
 <property name="rootClass" value=""/> -->
 <!\-\- 参考javaClientGenerator 的  rootInterface 属性 <property name="rootInterface" value=
"/> -->
 <!\-\- 如果设置了runtimeCatalog，那么在生成的SQL中，使用该指定的catalog，而不是table元
上的catalog <property name="runtimeCatalog" value=""/> -->
 <!\-\- 如果设置了runtimeSchema，那么在生成的SQL中，使用该指定的schema，而不是table元
上的schema <property name="runtimeSchema" value=""/> -->
 <!\-\- 如果设置了runtimeTableName，那么在生成的SQL中，使用该指定的tablename，而不是tab
e元素上的tablename <property name="runtimeTableName" value=""/> -->
 <!\-\- 注意，该属性只针对MyBatis3Simple有用； 如果选择的runtime是MyBatis3Simple，那么
生成一个SelectAll方法，如果指定了selectAllOrderByClause，那么会在该SQL中添加指定的这个ord
r条件；
```

<property name="selectAllOrderByClause" value="age desc,username asc"/>-->
<!\-\- 如果设置为true，生成的model类会直接使用column本身的名字，而不会再使用驼峰命名方，比如BORN_DATE，生成的属性名字就是BORN_DATE,而不会是bornDate --> <property nam="useActualColumnNames" value="false"/>

<!\-\- generatedKey用于生成生成主键的方法，
如果设置了该元素，MBG会在生成的<insert>元素中生成一条正确的<selectKey>元素，该元素可选 column:主键的列名； sqlStatement：要生成的selectKey语句，有以下可选项： Cloudscape:相当selectKey的SQL为： VALUES IDENTITY_VAL_LOCAL() DB2 :相当于selectKey的SQL为： VAUES IDENTITY_VAL_LOCAL() DB2_MF :相当于selectKey的SQL为：SELECT IDENTITY_VAL_LCAL() FROM SYSIBM.SYSDUMMY1 Derby :相当于selectKey的SQL为：VALUES IDENTITY_VAL_LOCAL() HSQLDB :相当于selectKey的SQL为：CALL IDENTITY() Informix :相当于selectKey的SL为：select dbinfo('sqlca.sqlerrd1') from systables where tabid=1 MySql :相当于selectKey的QL为：SELECT LAST_INSERT_ID() SqlServer :相当于selectKey的SQL为：SELECT SCOPE_IDENTIY() SYBASE :相当于selectKey的SQL为：SELECT @@IDENTITY JDBC :相当于在生成的insert素上添加useGeneratedKeys="true"和keyProperty属性 <generatedKey column="id" sqlStatemnt="JDBC"/> -->
<!-- 该元素会在根据表中列名计算对象属性名之前先重命名列名，非常适合用于表中的列都有公用前缀字符串的时候， 比如列名为：CUST_ID,CUST_NAME,CUST_EMAIL,CUST_ADDRESS等；么就可以设置searchString为"^CUST_"，并使用空白替换，那么生成的Customer对象中的属性名称不是 custId,custName等，而是先被替换为ID,NAME,EMAIL,然后变成属性：id，name，email；
注意，MBG是使用java.util.regex.Matcher.replaceAll来替换searchString和replaceString的， 如使用了columnOverride元素，该属性无效；
<columnRenamingRule searchString="" replaceString=""/> -->

<!\-\- 用来修改表中某个列的属性，MBG会使用修改后的列来生成domain的属性； column:要重设置的列名； 注意，一个table元素中可以有多个columnOverride元素哈~ --> <!\-\- <columnOverride column="username">--> <!\-\- 使用property属性来指定列要生成的属性名称 --> <!\-\- <poperty name="property" value="userName"/>-->
<!\-\- javaType用于指定生成的domain的属性类型，使用类型的全限定名 <property name="javaype" value=""/> -->
<!\-\- jdbcType用于指定该列的JDBC类型 <property name="jdbcType" value=""/> -->
<!\-\- typeHandler 用于指定该列使用到的TypeHandler，如果要指定，配置类型处理器的全限定名注意，mybatis中，不会生成到mybatis-config.xml中的typeHandler 只会生成类似：where id = #{d,jdbcType=BIGINT,typeHandler=com._520it.mybatis.MyTypeHandler}的参数描述 <property nme="jdbcType" value=""/> -->
<!\-\- 参考table元素的delimitAllColumns配置，默认为false <property name="delimitedColumName" value=""/> --> <!--</columnOverride>-->
<!\-\- ignoreColumn设置一个MGB忽略的列，如果设置了该列，那么在生成的domain中，生成的QL中，都不会有该列出现 column:指定要忽略的列的名字； delimitedColumnName：参考table元的delimitAllColumns配置，默认为false
注意，一个table元素中可以有多个ignoreColumn元素 <ignoreColumn column="deptId" delimitdColumnName=""/> --> </table>

</context>
</generatorConfiguration>

userMapper.java

List<User> selectByExample(UserCriteria example);

userMapper.xml

```
<select id="selectByExample" parameterType="com.zkjkgc.hpeis.entity.UserCriteria" resultM
p="BaseResultMap">
  select
  <if test="distinct">
  distinct
  </if>
 <include refid="Base\_Column\_List" />
  from SYS_USER
  <if test="_parameter != null">
 <include refid="Example\_Where\_Clause" />
 </if>  <if test="orderByClause != null">
  order by ${orderByClause}
  </if>
</select>
```

UserServiceImpl.java

```
@Override
public Map selectByExampleWithRowbounds(HttpServletRequest request) {
    Map map = new HashMap();
 int pageSize = Integer.parseInt(request.getParameter("rows"));
 int pageNum = Integer.parseInt(request.getParameter("page"));

  PageHelper.startPage(pageNum,pageSize);
  List<User> list = userMapper.selectByExample(null);

  PageInfo pageInfo = new PageInfo(list);
  map.put("rows",pageInfo.getList());
  map.put("total",pageInfo.getTotal());
 return map;
}
```