

SSM中MyBatis分页插件-PageHelper

一、PageHelper概述

开源的Mybatis分页插件

github地址:<https://github.com/pagehelper/Mybatis-PageHelper>

核心原理:通过ThreadLocal来存放分页信息

原理总结:PageHelper首先将前台传递的参数保存到page对象中，然后将page的副本放入ThreadLocal中，这样保证分页的时候，参数互不影响，最后利用了mybatis提供的拦截器，得到ThreadLocal的值，重新拼装分页SQL，完成分页。

二、分页基本步骤

1、pom.xml添加依赖

```
1 <!--mybatis分页插件 -->
2 <dependency>
3     <groupId>com.github.pagehelper</groupId>
4     <artifactId>pagehelper</artifactId>
5     <version>5.1.11</version>
6 </dependency>
```

2、配置插件

```
1 <!-- 在MyBatis配置文件中配置分页插件 -->
2 <plugins>
3     <plugin interceptor="com.github.pagehelper.PageInterceptor">
4         <!-- 设置数据库类型 Oracle,MySQL,MariaDB,SQLite,Hsqldb,PostgreSQL六种数据库方言-->
5         <property name="helperDialect" value="mysql"/>
6     </plugin>
7 </plugins>
```

3、SSM整合配置文件的编写【略】

4、定义项目结构和代码

com.ssm.entity|com.ssm.dao|com.ssm.service| com.ssm.service.impl|com.ssm.controller

5、配置web.xml

6、测试功能

三、分页代码展示

根据学生名字查询所有学生

A、导入依赖

```
1 <dependencies>
2   <!--mybatis依赖-->
3   <dependency>
4     <groupId>org.mybatis</groupId>
5     <artifactId>mybatis</artifactId>
6     <version>3.5.4</version>
7   </dependency>
8   <!--spring-web依赖-->
9   <dependency>
10    <groupId>org.springframework</groupId>
11    <artifactId>spring-webmvc</artifactId>
12    <version>5.2.4.RELEASE</version>
13  </dependency>
14  <!--mybatis-spring依赖-->
15  <dependency>
16    <groupId>org.mybatis</groupId>
17    <artifactId>mybatis-spring</artifactId>
18    <version>2.0.4</version>
19  </dependency>
20  <!--bonecp连接池配置 -->
21  <dependency>
22    <groupId>com.jolbox</groupId>
23    <artifactId>bonecp</artifactId>
24    <version>0.8.0.RELEASE</version>
25  </dependency>
26  <!--mysql驱动-->
27  <dependency>
28    <groupId>mysql</groupId>
29    <artifactId>mysql-connector-java</artifactId>
30    <version>5.1.48</version>
31  </dependency>
32  <!--spring-jdbc驱动-->
33  <dependency>
34    <groupId>org.springframework</groupId>
35    <artifactId>spring-jdbc</artifactId>
36    <version>5.2.4.RELEASE</version>
37  </dependency>
38  <!--加入日志log4j -->
39  <dependency>
40    <groupId>log4j</groupId>
```

```

41     <artifactId>log4j</artifactId>
42     <version>1.2.17</version>
43 </dependency>
44 <!--slf4j-nop依赖-->
45 <!--导入jstl依赖 -->
46 <dependency>
47     <groupId>javax.servlet</groupId>
48     <artifactId>jstl</artifactId>
49     <version>1.2</version>
50 </dependency>
51 <!--mybatis分页插件 -->
52 <dependency>
53     <groupId>com.github.pagehelper</groupId>
54     <artifactId>pagehelper</artifactId>
55     <version>5.1.11</version>
56 </dependency>
57 </dependencies>

```

B、配置文件

a、spring-config.xml

```

1  <?xml version="1.0" encoding="UTF-8" ?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xmlns:p="http://www.springframework.org/schema/p"
5      xmlns:c="http://www.springframework.org/schema/c"
6      xmlns:aop="http://www.springframework.org/schema/aop"
7      xmlns:context="http://www.springframework.org/schema/context"
8      xmlns:tx="http://www.springframework.org/schema/tx"
9      xsi:schemaLocation="http://www.springframework.org/schema/beans
10         https://www.springframework.org/schema/beans/spring-beans.xsd
11         http://www.springframework.org/schema/aop
12         https://www.springframework.org/schema/aop/spring-aop.xsd
13         http://www.springframework.org/schema/context
14         https://www.springframework.org/schema/context/spring-context.xsd
15         http://www.springframework.org/schema/tx
16         https://www.springframework.org/schema/tx/spring-tx.xsd">
17
18  <!--配置数据源-->
19  <bean id="dataSource" class="com.jolbox.bonecp.BoneCPDataSource"
20      p:driverClass="com.mysql.jdbc.Driver" p:jdbcUrl="jdbc:mysql://localhost:3306/ms"
21      p:username="root" p:password="admin"/>
22
23  <!--配置SqlSessionFactory对象 -->
24  <bean id="sqlFactroy" class="org.mybatis.spring.SqlSessionFactoryBean"
25      p:dataSource-ref="dataSource" p:configLocation="classpath:mybatis-config.xml"/>
26
27  <!--注入映射器-->
28  <!--basePackage指定了扫描的基准包，批量产生映射器实现类 -->
29  <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer"
30      p:basePackage="com.ssm.dao"/>

```

```

31
32      <!--配置基于注解扫描的Bean 将pojo实例化到spring容器中-->
33      <context:component-scan base-package="com.ssm"/>
34
35      <!--配置事务管理器 -->
36      <bean id="txManager"
37      class="org.springframework.jdbc.datasource.DataSourceTransactionManager"
38      p:dataSource-ref="dataSource"/>
39
40      <!--配置使用注解声明式事务处理-->
41      <tx:annotation-driven transaction-manager="txManager"/>
42  </beans>

```

b、springmvc-config.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xmlns:p="http://www.springframework.org/schema/p"
5      xmlns:mvc="http://www.springframework.org/schema/mvc"
6      xmlns:context="http://www.springframework.org/schema/context"
7      xsi:schemaLocation="http://www.springframework.org/schema/mvc
8          http://www.springframework.org/schema/mvc/spring-mvc.xsd
9          http://www.springframework.org/schema/context
10             http://www.springframework.org/schema/context/spring-context.xsd
11             http://www.springframework.org/schema/beans
12             http://www.springframework.org/schema/beans/spring-beans.xsd">
13      <!--扫描组件 -->
14      <!--告诉Spring 到哪里去找标记为@Controller 的Controller控制器-->
15      <context:component-scan base-package="com.ssm.controller"/>
16      <!--启用注解驱动-mvc配置 -->
17      <mvc:annotation-driven/>
18      <!--配置视图解析器 -->
19      <bean id="viewResolver"
20      class="org.springframework.web.servlet.view.InternalResourceViewResolver"
21      p:prefix="/WEB-INF/jsp/" p:suffix=".jsp"/>
22
23      <!--添加multipartResolver的bean支持文件上传 maxUploadSize 上传文件的最大大小，单位为字节-->
24      <!-- <bean id="multipartResolver"
25      class="org.springframework.web.multipart.commons.CommonsMultipartResolver"
26      p:maxUploadSize="802400" p:defaultEncoding="utf-8"/> -->
27  </beans>

```

c、mybatis-config.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE configuration
3      PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
4      "http://mybatis.org/dtd/mybatis-3-config.dtd">
5  <configuration>
6      <!--配置日志实现 -->
7      <settings>

```

```

8      <setting name="logImpl" value="LOG4J"/>
9  </settings>
10 <!--给实体类取别名-->
11 <typeAliases>
12     <!--批量取别名 默认为实体类首字母小写 -->
13     <package name="com.ssm.entity"/>
14 </typeAliases>
15
16 <!-- 配置分页插件 -->
17 <plugins>
18     <plugin interceptor="com.github.pagehelper.PageInterceptor">
19         <!-- 设置数据库类型 Oracle,MySQL,MariaDB,SQLite,Hsqldb,PostgreSQL六种数据库方言-->
20         <property name="helperDialect" value="mysql"/>
21     </plugin>
22 </plugins>
23
24 <!--映射器-告诉MyBatis去哪里找到sql映射文件 -->
25 <mappers>
26     <!--批量加载映射文件 -->
27     <package name="com.ssm.dao"/>
28 </mappers>
29 </configuration>

```

d、log4j.properties

```

1  ### \u8BBE\u7F6ELogger\u8F93\u51FA\u7EA7\u522B\u548C\u8F93\u51FA\u76EE\u7684\u5730
   ###
2  log4j.rootLogger=debug,stdout,logfile
3
4  ### \u628A\u65E5\u5FD7\u4FE1\u606F\u8F93\u51FA\u5230\u63A7\u5236\u53F0 ###
5  log4j.appender.stdout=org.apache.log4j.ConsoleAppender
6  log4j.appender.stdout.Target=System.out
7  log4j.appender.stdout.layout=org.apache.log4j.SimpleLayout
8
9  ### \u628A\u65E5\u5FD7\u4FE1\u606F\u8F93\u51FA\u5230\u6587\u4EF6\uFF1Ajbit.log ###
10 log4j.appender.logfile=org.apache.log4j.FileAppender
11 log4j.appender.logfile.File=jbit.log
12 log4j.appender.logfile.layout=org.apache.log4j.PatternLayout
13 log4j.appender.logfile.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss}%l %F %p %m%n

```

C、代码编写

a、实体类编写

```

1  public class Student {
2
3      private Integer sno;
4      private String pwd;
5      private String sname;
6      private String sex;
7      private Integer age;
8      private Integer gid;

```

```
9     private String phone;
10    private String address;
11    private String picpath; //图片地址
12
13    public Integer getSno() {
14        return sno;
15    }
16    public void setSno(Integer sno) {
17        this.sno = sno;
18    }
19    public String getPwd() {
20        return pwd;
21    }
22    public void setPwd(String pwd) {
23        this.pwd = pwd;
24    }
25    public String getSname() {
26        return sname;
27    }
28    public void setSname(String sname) {
29        this.sname = sname;
30    }
31    public String getSex() {
32        return sex;
33    }
34    public void setSex(String sex) {
35        this.sex = sex;
36    }
37    public Integer getAge() {
38        return age;
39    }
40    public void setAge(Integer age) {
41        this.age = age;
42    }
43    public Integer getGid() {
44        return gid;
45    }
46    public void setGid(Integer gid) {
47        this.gid = gid;
48    }
49    public String getPhone() {
50        return phone;
51    }
52    public void setPhone(String phone) {
53        this.phone = phone;
54    }
55    public String getAddress() {
56        return address;
57    }
58    public void setAddress(String address) {
59        this.address = address;
60    }
61    public String getPicpath() {
```

```

62         return picpath;
63     }
64     public void setPicpath(String picpath) {
65         this.picpath = picpath;
66     }
67 }

```

b、数据访问层

```

1 public interface StudentMapper {
2     List<Student> getStulistByMany(@Param("sname")String sname);
3 }

```

c、映射文件

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE mapper
3     PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
4     "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
5 <mapper namespace="com.ssm.dao.StudentMapper">
6
7     <!--根据多条件查询学生 -->
8     <select id="getStulistByMany" resultType="student">
9         select * from student
10        <trim prefix="where" prefixOverrides="and | or">
11            <if test="sname!=null and sname!=''">and sname like concat('%',{sname},'%')
12        </if>
13        </trim>
14    </select>
15 </mapper>

```

d、业务逻辑层

```

1 public interface StudentService { //业务接口
2
3     List<Student> getStulistByMany(@Param("sname")String sname);
4 }
5 @Service
6 public class StudentServiceImpl implements StudentService { //业务接口实现类
7
8     @Autowired //或者@Resource进行自动装配
9     private StudentMapper sMapper;
10
11     @Override
12     public List<Student> getStulistByMany(String sname) {
13         return sMapper.getStulistByMany(sname);
14     }
15 }

```

d、控制层

```

1  @Controller
2  public class StudentController {
3      @Autowired
4      private StudentService stuService;
5
6      @RequestMapping("/stulist.do")
7      public ModelAndView selectListByMany(
8          @RequestParam(required=true,defaultValue="1")Integer page
9          ,@RequestParam(required=true,defaultValue="")String sname){
10         ModelAndView mv=new ModelAndView(); //声明一个ModelAndView对象
11         PageHelper.startPage(page,3); //开始分页
12         PageHelper.orderBy("age desc"); //排序规则
13         List<Student> list = stuService.getStulistByMany(sname);
14         PageInfo<Student> p=new PageInfo<Student>(list);
15         mv.addObject("page",p);
16         mv.addObject("slist",list);
17         mv.addObject("sname",sname);
18         mv.setViewName("index");
19         return mv;
20     }
21 }

```

e、首页列表

```

1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2      pageEncoding="UTF-8"%>
3  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
4  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
5      "http://www.w3.org/TR/html4/loose.dtd">
6  <html>
7  <head>
8      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
9      <title>首页</title>
10     <script type="text/javascript" src="js/jquery-2.1.4.min.js"></script>
11     <script type="text/javascript">
12         $(function(){
13             $(".base:even").css("background-color","green");
14         });
15     </script>
16 </head>
17 <body>
18     <div align="center">
19         <form action="stulist.do" method="post">
20             姓名:<input type="text" name="sname" value="${sname}"/>
21             <input type="submit" value="查询"/>
22         </form>
23         <table align="center" border="1" cellpadding="0" cellspacing="0" width="50%">
24             <tr>
25                 <th>学号</th>
26                 <th>姓名</th>
27                 <th>性别</th>
28                 <th>年龄</th>

```



```

28         <th>操作</th>
29     </tr>
30     <c:forEach items="${slist}" var="stu">
31         <tr class="base">
32             <th>${stu.sno}</th>
33             <th>${stu.sname}</th>
34             <th>${stu.sex}</th>
35             <th>${stu.age}</th>
36             <th><a href="toDetail.do?sno=${stu.sno}">详情</a>|
37                 <a href="toUpdate.do?sno=${stu.sno}">修改</a>|
38                 <a href="deleteStu.do?sno=${stu.sno}">删除</a></th>
39         </tr>
40     </c:forEach>
41 </table>
42 <p>
43     <c:choose>
44         <c:when test="${page.isFirstPage==true}">首页
45             <a href="stulist.do?page=${page.nextPage}&&sname=${sname}">下页</a>
46             <a href="stulist.do?page=${page.navigateLastPage}&&sname=${sname}">末页</a>
47         </c:when>
48         <c:when test="${page.isLastPage==true}">
49 <a href="stulist.do?page=${page.navigateFirstPage}&&sname=${sname}">首页</a>
50 <a href="stulist.do?page=${page.prePage}&&sname=${sname}">上页</a>&nbsp;末页
51         </c:when>
52         <c:otherwise>
53 <a href="stulist.do?page=${page.navigateFirstPage}&&sname=${sname}">首页</a>
54 <a href="stulist.do?page=${page.prePage}&&sname=${sname}">上页</a>
55 <a href="stulist.do?page=${page.nextPage}&&sname=${sname}">下页</a>
56 <a href="stulist.do?page=${page.navigateLastPage}&&sname=${sname}">末页</a>
57         </c:otherwise>
58     </c:choose>
59     当前页${page.size}条，总共${page.total}条，第${page.pageNum}/${page.pages}页
60 </p>
61 </div>
62 </body>
63 </html>

```

D、配置web.xml

```

1 <context-param>
2     <param-name>contextConfigLocation</param-name>
3     <param-value>classpath:spring-config.xml</param-value>
4 </context-param>
5 <listener>
6     <listener-class>org.springframework.web.context.ContextLoaderListener</listener-
class>
7 </listener>
8 <servlet>
9     <servlet-name>smvc</servlet-name>
10    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
11    <init-param>
12        <param-name>contextConfigLocation</param-name>

```

```
13     <param-value>classpath:springmvc-config.xml</param-value>
14     </init-param>
15     <load-on-startup>1</load-on-startup>
16 </servlet>
17 <servlet-mapping>
18     <servlet-name>smvc</servlet-name>
19     <url-pattern>*.do</url-pattern>
20 </servlet-mapping>
21 <filter>
22     <filter-name>encoding</filter-name>
23     <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-
class>
24     <init-param>
25         <param-name>encoding</param-name>
26         <param-value>utf-8</param-value>
27     </init-param>
28 </filter>
29 <filter-mapping>
30     <filter-name>encoding</filter-name>
31     <url-pattern>/*</url-pattern>
32 </filter-mapping>
```

E、测试功能

1 | <http://localhost:8080/07-SSM-MyBatisPage/stulist.do>

四、课堂总结

五、更多视频扩展资源

今日头条：<https://www.toutiao.com/c/user/1736832959129566/> 进去后-选择视频-在线点击观看

哔哩哔哩：<https://space.bilibili.com/425233867>