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的副本存放入ThreadLoacl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

```
<!--mybatis分页插件 -->
<dependency>
    <groupId>com.github.pagehelper</groupId>
    <artifactId>pagehelper</artifactId>
    <version>5.1.11</version>
</dependency>
```

2、配置插件

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

加载spring-config.xml | springmvc-config.xml | 中文乱码

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

```
<dependencies>
    <!--mybatis依赖-->
    <dependency>
        <groupId>org.mybatis</groupId>
        <artifactId>mybatis</artifactId>
        <version>3.5.4</version>
    </dependency>
    <!--spring-web依赖-->
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-webmvc</artifactId>
        <version>5.2.4.RELEASE</version>
    </dependency>
    <!--mybatis-spring依赖-->
    <dependency>
        <groupId>org.mybatis</groupId>
        <artifactId>mybatis-spring</artifactId>
        <version>2.0.4</version>
    </dependency>
    <!--bonecp连接池配置 -->
    <dependency>
        <groupId>com.jolbox</groupId>
        <artifactId>bonecp</artifactId>
        <version>0.8.0.RELEASE</version>
    </dependency>
    <!--mysql驱动-->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>5.1.48</version>
    </dependency>
    <!--spring-jdbc驱动-->
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-jdbc</artifactId>
        <version>5.2.4.RELEASE</version>
    </dependency>
     <!--加入日志log4j -->
     <dependency>
        <groupId>log4j</groupId>
        <artifactId>log4j</artifactId>
        <version>1.2.17</version>
    </dependency>
    <!--slf4j-nop依赖-->
    <!--导入jstl依赖 -->
    <dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>jstl</artifactId>
        <version>1.2</version>
    </dependency>
    <!--mybatis分页插件 -->
    <dependency>
      <groupId>com.github.pagehelper</groupId>
      <artifactId>pagehelper</artifactId>
      <version>5.1.11</version>
    </dependency>
  </dependencies>
```

B、配置文件

a、spring-config.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:c="http://www.springframework.org/schema/c"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        https://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/aop
        https://www.springframework.org/schema/aop/spring-aop.xsd
        http://www.springframework.org/schema/context
        https://www.springframework.org/schema/context/spring-context.xsd
        http://www.springframework.org/schema/tx
        https://www.springframework.org/schema/tx/spring-tx.xsd">

   <!--配置数据源-->
   <bean id="dataSource" class="com.jolbox.bonecp.BoneCPDataSource"
   p:driverClass="com.mysql.jdbc.Driver" p:jdbcUrl="jdbc:mysql://localhost:3306/ms"
   p:username="root" p:password="admin"/>

   <!--配置SqlSeesionFactory对象 -->
   <bean id="sqlFactroy" class="org.mybatis.spring.SqlSessionFactoryBean"
    p:dataSource-ref="dataSource" p:configLocation="classpath:mybatis-config.xml"/>

    <!--注入映射器-->
    <!--basePackage指定了扫描的基准包，批量产生映射器实现类 -->
    <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer"
    p:basePackage="com.ssm.dao"/>
```

```xml

    <!--配置基于注解扫描的Bean 将pojo实例化到spring容器中-->
    <context:component-scan base-package="com.ssm"/>

    <!--配置事务管理器 -->
    <bean id="txManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager"
    p:dataSource-ref="dataSource"/>

    <!--配置使用注解声明式事务处理-->
    <tx:annotation-driven transaction-manager="txManager"/>
 </beans>
```

b、springmvc-config.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:mvc="http://www.springframework.org/schema/mvc"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/mvc
    http://www.springframework.org/schema/mvc/spring-mvc.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context.xsd
        http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd">
    <!--扫描组件 -->
    <!--告诉Spring 到哪里去找标记为@Controller 的Controller控制器-->
    <context:component-scan base-package="com.ssm.controller"/>
    <!--启用注解驱动-mvc配置 -->
    <mvc:annotation-driven/>
    <!--配置视图解析器 -->
    <bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver"
    p:prefix="/WEB-INF/jsp/" p:suffix=".jsp"/>

  <!--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-->
  <!-- <bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver"
  p:maxUploadSize="802400" p:defaultEncoding="utf-8"/> -->
</beans>
```

c、mybatis-config.xml

```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>
  <!--配置日志实现 -->
  <settings>
```

```
    <setting name="logImpl" value="LOG4J"/>
  </settings>
   <!--给实体类取别名-->
   <typeAliases>
      <!--批量取别名 默认为实体类首字母小写 -->
      <package name="com.ssm.entity"/>
   </typeAliases>

    <!-- 配置分页插件 -->
    <plugins>
       <plugin interceptor="com.github.pagehelper.PageInterceptor">
      <!-- 设置数据库类型 Oracle,Mysql,MariaDB,SQLite,Hsqldb,PostgreSQL六种数据库方言-->
         <property name="helperDialect" value="mysql"/>
      </plugin>
      </plugins>

    <!--映射器-告诉MyBatis去哪里找到sql映射文件 -->
    <mappers>
        <!--批量加载映射文件 -->
        <package name="com.ssm.dao"/>
    </mappers>
</configuration>
```

d、log4j.properties

```
### \u8BBE\u7F6ELogger\u8F93\u51FA\u7EA7\u522B\u548C\u8F93\u51FA\u76EE\u7684\u5730 ###
log4j.rootLogger=debug,stdout,logfile

### \u628A\u65E5\u5FD7\u4FE1\u606F\u8F93\u51FA\u5230\u63A7\u5236\u53F0 ###
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.SimpleLayout

### \u628A\u65E5\u5FD7\u4FE1\u606F\u8F93\u51FA\u5230\u6587\u4EF6\uFF1Ajbit.log ###
log4j.appender.logfile=org.apache.log4j.FileAppender
log4j.appender.logfile.File=jbit.log
log4j.appender.logfile.layout=org.apache.log4j.PatternLayout
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

```
public class Student {

    private Integer sno;
    private String pwd;
    private String sname;
    private String sex;
    private Integer age;
    private Integer gid;
    private String phone;
    private String address;
    private String picpath; //图片地址

    public Integer getSno() {
        return sno;
    }
    public void setSno(Integer sno) {
        this.sno = sno;
    }
    public String getPwd() {
        return pwd;
    }
    public void setPwd(String pwd) {
        this.pwd = pwd;
    }
    public String getSname() {
        return sname;
    }
    public void setSname(String sname) {
        this.sname = sname;
    }
    public String getSex() {
        return sex;
    }
    public void setSex(String sex) {
        this.sex = sex;
    }
    public Integer getAge() {
        return age;
    }
    public void setAge(Integer age) {
        this.age = age;
    }
    public Integer getGid() {
        return gid;
    }
    public void setGid(Integer gid) {
        this.gid = gid;
    }
    public String getPhone() {
        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String getPicpath() {
        return picpath;
    }
    public void setPicpath(String picpath) {
        this.picpath = picpath;
    }
}
```

b、数据访问层

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

c、映射文件

```
<?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.ssm.dao.StudentMapper">

    <!--根据多条件查询学生 -->
    <select id="getStulistByMany" resultType="student">
      select * from student
      <trim prefix="where" prefixOverrides="and | or">
        <if test="sname!=null and sname!=''">and sname like concat('%',#{sname},'%')</if>
      </trim>
    </select>

</mapper>
```

d、业务逻辑层

```
public interface StudentService { //业务接口

   List<Student> getStulistByMany(@Param("sname")String sname);
}
@Service
public class StudentServiceImpl implements StudentService {//业务接口实现类

    @Autowired //或者@Resource进行自动装配
    private StudentMapper sMapper;

    @Override
    public List<Student> getStulistByMany(String sname) {
        return sMapper.getStulistByMany(sname);
    }
}
```

d、控制层

```
@Controller
public class StudentController {
    @Autowired
    private StudentService stuService;

    @RequestMapping("/stulist.do")
    public ModelAndView selectListByMany(
            @RequestParam(required=true,defaultValue="1")Integer page
            ,@RequestParam(required=true,defaultValue="")String sname){
        ModelAndView mv=new ModelAndView(); //声明一个ModelAndView对象
        PageHelper.startPage(page,3); //开始分页
        PageHelper.orderBy("age desc"); //排序规则
        List<Student> list = stuService.getStulistByMany(sname);
        PageInfo<Student> p=new PageInfo<Student>(list);
            mv.addObject("page",p);
            mv.addObject("slist",list);
            mv.addObject("sname",sname);
            mv.setViewName("index");
            return mv;
    }
}
```

e、首页列表

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>首页</title>
<script type="text/javascript" src="js/jquery-2.1.4.min.js"></script>
<script type="text/javascript">
$(function(){
    $(".base:even").css("background-color","green");
});
</script>
</head>
<body>
<div align="center">
     <form action="stulist.do" method="post">
            姓名:<input type="text" name="sname" value="${sname}"/>
       <input type="submit" value="查询"/>
      </form>
    <table align="center" border="1" cellpadding="0" cellspacing="0" width="50%">
        <tr>
            <th>学号</th>
            <th>姓名</th>
            <th>性别</th>
            <th>年龄</th>
            <th>操作</th>
        </tr>
        <c:forEach items="${slist}" var="stu">
            <tr class="base">
                <th>${stu.sno}</th>
                <th>${stu.sname}</th>
                <th>${stu.sex}</th>
                <th>${stu.age}</th>
                <th><a href="toDetail.do?sno=${stu.sno}">详情</a>|
                    <a href="toUpdate.do?sno=${stu.sno}">修改</a>|
                    <a href="deleteStu.do?sno=${stu.sno}">删去</a></th>
            </tr>
        </c:forEach>
    </table>
      <p>
      <c:choose>
        <c:when test="${page.isFirstPage==true}">首页
          <a href="stulist.do?page=${page.nextPage}&&sname=${sname}">下页</a>
         <a href="stulist.do?page=${page.navigateLastPage}&&sname=${sname}">末页</a>
        </c:when>
        <c:when test="${page.isLastPage==true}">
     <a href="stulist.do?page=${page.navigateFirstPage}&&sname=${sname}">首页</a>
      <a href="stulist.do?page=${page.prePage}&&sname=${sname}">上页</a>&nbsp;末页
        </c:when>
        <c:otherwise>
        <a href="stulist.do?page=${page.navigateFirstPage}&&sname=${sname}">首页</a>
        <a href="stulist.do?page=${page.prePage}&&sname=${sname}">上页</a>
        <a href="stulist.do?page=${page.nextPage}&&sname=${sname}">下页</a>
        <a href="stulist.do?page=${page.navigateLastPage}&&sname=${sname}">末页</a>
        </c:otherwise>
      </c:choose>
        当前页${page.size}条，总共${page.total}条，第${page.pageNum}/${page.pages}页
   </p>
</div>
</body>
</html>
```

D、配置web.xml

```
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:spring-config.xml</param-value>
  </context-param>
  <listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
  </listener>
  <servlet>
    <servlet-name>smvc</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param>
      <param-name>contextConfigLocation</param-name>
      <param-value>classpath:springmvc-config.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>smvc</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>
  <filter>
    <filter-name>encoding</filter-name>
    <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
    <init-param>
      <param-name>encoding</param-name>
      <param-value>utf-8</param-value>
    </init-param>
  </filter>
  <filter-mapping>
    <filter-name>encoding</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
```

E、测试功能

```
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