



链滴

后端 | SpringBoot2.5 整合 MyBatis 分页

作者: [z875479694h](#)

原文链接: <https://ld246.com/article/1627629387661>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



一.前言

这是一个基于SpringBoot2.5.3整合MyBatis3.5.7使用PageHelper实现分页的极简教程，笔者使用到技术及版本如下：

1. SpringBoot 2.5.3
2. Mybatis 3.5.7
3. PageHelper 5.2.1

写博客的起因是某位程序猿小姐姐在使用低代码平台时，跑通了数据查询功能后尝试对数据进行分页现不起作用，所以笔者花了1个小时快速实现了整合demo供小姐姐研究学习。

二.pom文件

pom文件核心部分

```
<!--SpringBoot的Web依赖-->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<!--Mybatis的SpringBoot版本的依赖-->
<dependency>
    <groupId>org.mybatis.spring.boot</groupId>
    <artifactId>mybatis-spring-boot-starter</artifactId>
    <version>2.2.0</version>
</dependency>
<!--PageHelper分页插件的SpringBoot版本的依赖-->
<dependency>
```

```

        <groupId>com.github.pagehelper</groupId>
        <artifactId>pagehelper-spring-boot-starter</artifactId>
        <version>1.3.1</version>
    </dependency>
<!--mysql驱动-->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <scope>runtime</scope>
    </dependency>
<!--lombok工具-->
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <optional>>true</optional>
    </dependency>

```

pom文件全文

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/
MLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xs
/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.5.3</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>xyz.hcworld.mybatis</groupId>
    <artifactId>pagedemo</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>pagedemo</name>
    <description>Mybatis分页</description>
    <properties>
        <java.version>1.8</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.mybatis.spring.boot</groupId>
            <artifactId>mybatis-spring-boot-starter</artifactId>
            <version>2.2.0</version>
        </dependency>

        <dependency>
            <groupId>com.github.pagehelper</groupId>
            <artifactId>pagehelper-spring-boot-starter</artifactId>
            <version>1.3.1</version>

```

```

</dependency>

<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <scope>runtime</scope>
</dependency>
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <optional>true</optional>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>
</dependencies>

<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <configuration>
        <excludes>
          <exclude>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
          </exclude>
        </excludes>
      </configuration>
    </plugin>
  </plugins>
</build>

</project>

```

三. 配置文件（基于yml）

```

# 配置mysql地址连接属性等
spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://xxx.xxx.xxx.xxx:3306/test?useUnicode=true&useSSL=false&characterEncoding=utf8&serverTimezone=Asia/Shanghai
    username: xxx
    password: xxx
# Mybatis的模型地址和Mapper.xml文件地址
mybatis:
  type-aliases-package: xyz.hcworld.mybatis.pagedemo.entity
  mapper-locations: classpath*/mapper/**Mapper.xml
# 分页配置, 分页类型为mysql
pagehelper:

```

```
helper-dialect: mysql
# 分页合理化参数:pageNum<=0 时会查询第一页,pageNum>pages (超过总数时) , 会查询最后一页
reasonable: true
# 支持通过 Mapper 接口参数来传递分页参数 (下文第三种查询用到)
support-methods-arguments: true
```

四.模型及数据统一返回类

模型

```
package xyz.hcworld.mybatis.pagedemo.entity;

import lombok.Data;

import java.util.Date;

/**
 * @ClassName: UserInfo
 * @Author: 张红尘
 * @Date: 2021-07-29
 * @Version: 1.0
 */
@Data
public class UserInfo {

    /**
     * 用户编号
     */
    private int userId;
    /**
     * 用户姓名
     */
    private String userName;
    /**
     * 省份
     */
    private String province;
    /**
     * 注册时间
     */
    private Date createTime;
}
```

数据统一返回类型

```
package xyz.hcworld.mybatis.pagedemo.lang;

import lombok.Data;

import java.io.Serializable;
```

```

/**
 * 统一返回信息模板
 * @ClassName: Result
 * @Author: 张红尘
 * @Date: 2021-04-23
 * @Version: 1.0
 */
@Data
public class Result implements Serializable {
    /**
     * 0成功, -1失败
     */
    private int status;
    /**
     * 消息
     */
    private String msg;
    /**
     * 数据域
     */
    private Object data;
    /**
     * 跳转链接
     */
    private String action;

    /**
     * 成功 (默认msg+无data)
     * @return
     */
    public static Result success() {
        return Result.success("操作成功", null);
    }
    /**
     * 成功 (默认msg+自定义data)
     * @param data 数据域
     * @return
     */
    public static Result success(Object data) {
        return Result.success("操作成功", data);
    }
    /**
     * 成功 (自定义msg+自定义data)
     * @param msg 消息
     * @param data 数据域
     * @return
     */
    public static Result success(String msg, Object data) {
        return Result.success(msg, data,null);
    }
    /**
     * 成功 (自定义msg+自定义data+自定义路径)
     * @param msg 消息

```

```

* @param data 数据域
* @return
*/
public static Result success(String msg, Object data,String action) {
    Result result = new Result();
    result.status = 0;
    result.msg = msg;
    result.data = data;
    result.action = action;
    return result;
}

/**
* 失败
* @param msg 消息
* @return
*/
public static Result fail(String msg) {
    Result result = new Result();
    result.status = -1;
    result.msg = msg;
    result.data = null;
    return result;
}

public Result action(String action){
    this.action=action;
    return this;
}
}

```

五.Controller

```

package xyz.hcworld.mybatis.pagedemo.controller;

import com.github.pagehelper.PageHelper;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;
import xyz.hcworld.mybatis.pagedemo.entity.UserInfo;
import xyz.hcworld.mybatis.pagedemo.lang.Result;
import xyz.hcworld.mybatis.pagedemo.service.UserServer;

import java.util.ArrayList;
import java.util.Date;
import java.util.List;

/**
* @ClassName: UserController
* @Author: 张红尘

```

```

* @Date: 2021-07-29
* @Version: 1.0
*/
@RestController
public class UserController {
    @Autowired
    UserServer userServer;

    /**
     * 第一种分页方式
     * @param pageNum 页数
     * @param pageSize 每页多少条
     * @return
     */
    @GetMapping("/userinfo1")
    public Result userInfoList1(@RequestParam(defaultValue = "1") Integer pageNum, @Request
tParam(defaultValue = "10") Integer pageSize){
        PageHelper.startPage(pageNum,pageSize);
        return Result.success(userServer.userList());
    }

    /**
     * 第二种分页方式
     * @param pageNum 从第几条数据开始
     * @param pageSize 获取多少条多少条
     * @return
     */
    @GetMapping("/userinfo2")
    public Result userInfoList2(@RequestParam(defaultValue = "1") Integer pageNum,@Reque
tParam(defaultValue = "10") Integer pageSize){
        PageHelper.offsetPage(pageNum,pageSize);
        return Result.success(userServer.userList());
    }

    /**
     * 第三种分页方式
     * @param pageNum 页数
     * @param pageSize 每页多少条
     * @return
     */
    @GetMapping("/userinfo3")
    public Result userInfoList3(@RequestParam(defaultValue = "1") Integer pageNum,@Reque
tParam(defaultValue = "10") Integer pageSize){
        /**
         * 在mapper中加入@Param("pageNum") int pageNum,@Param("pageSize") int pageSize
         * 置要获取的页码和每页数据数
         */
        return Result.success(userServer.userList(pageNum,pageSize));
    }

    @GetMapping("/addUserinfo")
    public void addUserInfoList(){

```



```

        List<UserInfo> list = new ArrayList<>(101);
        for (int i = 0;i<100;i++){
            UserInfo userInfo = new UserInfo();
            userInfo.setUsername("张三第"+i+"号");
            userInfo.setProvince("广州xxx第"+i+"号");
            userInfo.setCreateTime(new Date());
            list.add(userInfo);
        }
        userServer.addUser(list);
    }
}
}

```

六.Server及其实现

```

package xyz.hcworld.mybatis.pagedemo.service;

import com.github.pagehelper.Page;
import xyz.hcworld.mybatis.pagedemo.entity.UserInfo;
import java.util.List;

/**
 * @ClassName: UserServer
 * @Author: 张红尘
 * @Date: 2021-07-29
 * @Version: 1.0
 */
public interface UserServer {
    /**
     * 查询用户
     * @return
     */
    Page<UserInfo> userList();

    /**
     * 分页查询用户
     * @return
     */
    Page<UserInfo> userList(Integer pageNum,Integer pageSize);

    /**
     * 新增用户
     */
    void addUser(List<UserInfo> list);
}

package xyz.hcworld.mybatis.pagedemo.service.impl;

```

```

import com.github.pagehelper.Page;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import xyz.hcworld.mybatis.pagedemo.entity.UserInfo;
import xyz.hcworld.mybatis.pagedemo.mapper.UserMapper;
import xyz.hcworld.mybatis.pagedemo.service.UserServer;

import java.util.List;

/**
 * @ClassName: UserServerImpl
 * @Author: 张红尘
 * @Date: 2021-07-29
 * @Version: 1.0
 */
@Service
public class UserServerImpl implements UserServer {
    @Autowired
    UserMapper userMapper;

    @Override
    public Page<UserInfo> userList() {
        return userMapper.findAllUser();
    }

    @Override
    public Page<UserInfo> userList(Integer pageNum, Integer pageSize) {
        return userMapper.findAllUser(pageNum,pageSize);
    }

    @Override
    public void addUser(List<UserInfo> list) {
        userMapper.addUserInfo(list);
    }
}

```

七.Mapper

```

package xyz.hcworld.mybatis.pagedemo.mapper;

import com.github.pagehelper.Page;
import org.apache.ibatis.annotations.Param;
import org.springframework.stereotype.Component;
import xyz.hcworld.mybatis.pagedemo.entity.UserInfo;

import java.util.List;

/**
 * @ClassName: UserMapper
 * @Author: 张红尘
 * @Date: 2021-07-29
 * @Version: 1.0
 */

```

```
@Component
public interface UserMapper {
```

```
/**
 * 查询全部用户（可手动设置按分页）
 * @return
 */
Page<UserInfo> findAllUser();
```

```
/**
 * 分页获取用户，第三种方法
 * @param pageNum
 * @param pageSize
 * @return
 */
Page<UserInfo> findAllUser(@Param("pageNum") Integer pageNum,
                          @Param("pageSize") Integer pageSize);
```

```
/**
 * 新增
 * @param userInfos
 */
void addUserInfo(List<UserInfo> userInfos);
```

```
}

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/ybatis-3-mapper.dtd">
<mapper namespace="xyz.hcworld.mybatis.pagedemo.mapper.UserMapper">
  <!-- 查询指定表的数据是否存在 -->
  <select id="findAllUser" resultType="xyz.hcworld.mybatis.pagedemo.entity.UserInfo" >
    SELECT *
    FROM user_info
  </select>
  <insert id="addUserInfo" parameterType="xyz.hcworld.mybatis.pagedemo.entity.UserInfo" >
    >
    INSERT INTO `test`.`user_info`
    (`userName`,
     `province`,
     `createTime`)
    VALUES
    <foreach collection="list" item="userInfo" separator=",">
      (#{userInfo.userName},
       #{userInfo.province},
       #{userInfo.createTime})
    </foreach >
  </insert>
</mapper>
```

八.启动文件

```
package xyz.hcworld.mybatis.pagedemo;
```

```
import org.mybatis.spring.annotation.MapperScan;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
@SpringBootApplication
```

```
//加了这个直接扫码整个包，就不需要给一个个Mapper文件加@Mapper标签了
```

```
@MapperScan(basePackages = "xyz.hcworld.mybatis.pagedemo.mapper")
```

```
public class PagedemoApplication {
```

```
    public static void main(String[] args) {
        SpringApplication.run(PagedemoApplication.class, args);
    }
}
```

九.三种分页方式的查询结果

第一种，根据页数以及数据条数查询当前页结果，可以看出，输入了页码即可获得数据。

注意 `pageHelper.startPage(m,n)`，只对最近的一次查询有效，就是设置完后，调用了Mapper的查方法后就失效了。

The first screenshot shows a REST client interface with the following details:

- Name: userinfo1
- Response:

```
{status: 0, msg: "操作成功",...}
action: null
data: [{userId: 1, userName: "张三第0号", province: "广州xxxx第0号", createTime: "2021-07-29T04:51:09.000+00:00"},...]
```

The second screenshot shows a REST client interface with the following details:

- Name: userinfo1?pageNum=2
- Response:

```
{status: 0, msg: "操作成功", data: [...], action: null}
action: null
data: [...]
```

第二种，指定第几条数据，并获取其后续的指定条数据，根据偏移量获取数据。（我都用这个了为啥自己手写分页，何必增加一个依赖，消耗服务器性能）

The screenshot shows a REST client interface with the following details:

- Name: userinfo2?pageNum=0
- Response:

```
{status: 0, msg: "操作成功",...}
action: null
data: [{"userId": 1, "userName": "张三第0号", "province": "广州xxxx第0号", "createTime": "2021-07-29T04:51:09.000+00:00"},...]
```

Name	X Headers	Preview	Response	Initiator	Timing
userinfo2?pageNum=0			<pre> (status: 0, msg: "操作成功",-) action: null data: [{userId: 3, userName: "张三第2号", province: "广州xxx第2号", createTime: "2021-07-29T04:51:09.000+00:00"},...] </pre>		
userinfo2?pageNum=2			<pre> (status: 0, msg: "操作成功",-) action: null data: [{userId: 3, userName: "张三第2号", province: "广州xxx第2号", createTime: "2021-07-29T04:51:09.000+00:00"},...] </pre>		

第三种，与第一种类似，只是把页码与数据量写到了Mapper中，

Name	X Headers	Preview	Response	Initiator	Timing
userinfo3?pageNum=1			<pre> (status: 0, msg: "操作成功",-) action: null data: [{userId: 1, userName: "张三第0号", province: "广州xxx第0号", createTime: "2021-07-29T04:51:09.000+00:00"},...] </pre>		

Name	X Headers	Preview	Response	Initiator	Timing
userinfo3?pageNum=1			<pre> (status: 0, msg: "操作成功", data: [,-], action: null) action: null data: [,-] </pre>		
userinfo3?pageNum=2			<pre> (status: 0, msg: "操作成功", data: [,-], action: null) action: null data: [,-] </pre>		

十.笔者的话

笔者最近比较忙，更新文章速度较慢，且急着给小姐姐解决问题，因此变量命名，方法命名等略有不规范，注释也没写到位请多多见谅。

十一.GitHub

GitHub:<https://github.com/z875479694h/mybatis-pagehelper-demo>