

# SpringBoot 整合 MyBatis-Plus

## 1. MyBatis-Plus 介绍

### 1.1 MyBatis-Plus 简介

MyBatis-Plus (简称 MP) 是一个 MyBatis 的**增强**工具，在 Mybatis 的基础上只做增强不做改变，为简化开发、提高效率而生(这是官方给的定义)。关于 mybatis-plus 的更多介绍及特性，可以参考 mybatis-plus 官网。

MyBatis 框架我们前面已经用过，虽然 MyBatis 可以直接在 xml 中通过 SQL 语句操作数据库，很是灵活。但因为其操作都要通过 SQL 语句进行，就必须写大量的 xml 文件，很麻烦。于是 MyBatis-Plus 应运而生，作为 MyBatis 的增强工具，使我们开发效率得到了质的飞跃。那么它是怎么增强的呢？其实就是它已经封装好了一些 crud 方法，我们不需要再写 xml 了，直接调用这些方法就行。

## 2. SpringBoot 中整合 MyBatis-Plus

### 2.1 SpringBoot 整合 Mybatis-Plus 入门案例

#### 2.1.1 导入数据库

导入“books 数据库创建语句-第 7 周.txt”数据库

#### 2.1.2 搭建工程

创建一个工程 springboot-mybatis-plus-bookstore

New Project

Server URL: [start.spring.io](https://start.spring.io)

Name:

Location:

Project will be created in: D:\Java\IdeaWorkspace\spring-book-mybatisplus-bookstore-2

☐ Create Git repository

Language: ☐ Java ☐ Kotlin ☐ Groovy

Type: ☐ Gradle - Groovy ☐ Gradle - Kotlin ☐ Maven

Group:

Artifact:

Package name:

JDK:

Java:

Packaging: ☐ Jar ☐ War

Next Cancel

New Project

Spring Boot:

☒ Download pre-built shared indexes for JDK and Maven libraries

Dependencies:

Search

- ☐ Spring Data JDBC
- ☐ Spring Data R2DBC
- ☐ MyBatis Framework
- ☐ Liquibase Migration
- ☐ Flyway Migration
- ☐ JOOQ Access Layer
- ☐ IBM DB2 Driver
- ☐ Apache Derby Database
- ☐ H2 Database
- ☐ HyperSQL Database
- ☐ MariaDB Driver
- ☐ MS SQL Server Driver
- ☒ MySQL Driver
- ☐ Oracle Driver
- ☐ PostgreSQL Driver

NoSQL

MySQL Driver

MySQL JDBC driver.

[Guide](#)

Added dependencies:

- × Spring Web
- × Thymeleaf
- × MySQL Driver

Previous Create Cancel

## 2.1.3 pom 文件

### 2.1.3.1 引入 MyBatis-Plus 依赖

**Spring-boot 3.2.4 版 匹配的 MyBatis-Plus 版本是 3.5.5**(Spring-boot 2 匹配的 MyBatis-Plus 版本是 3.5.3.1)

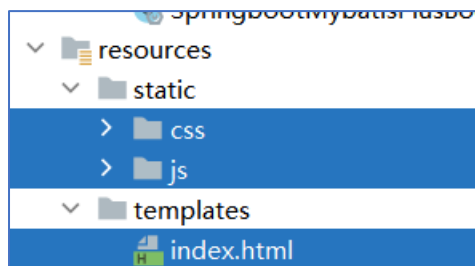
```
<!--mybatis-plus 的 springboot 支持 -->
<dependency>
    <groupId>com.baomidou</groupId>
    <artifactId>mybatis-plus-boot-starter</artifactId>
    <version>3.5.5</version>
</dependency>
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis-spring</artifactId>
    <version>3.0.3</version>
</dependency>
<!-- mybatis-plus 代码生成器 -->
<dependency>
    <groupId>com.baomidou</groupId>
    <artifactId>mybatis-plus-generator</artifactId>
    <version>3.5.5</version>
</dependency>

<!-- mybatisPlus Freemarker 模版引擎 -->
<dependency>
    <groupId>org.freemarker</groupId>
    <artifactId>freemarker</artifactId>
    <version>2.3.31</version>
</dependency>

<!-- 简化代码的工具包 -->
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
</dependency>
```

lombok 是一个工具，用于简化代码。

## 2.3 导入静态资源



## 3. 入门案例-CRUD 操作

### 3.1 创建 application.properties

在 src/main/resources 目录下创建 application.properties 文件

```
mybatis-plus.configuration.logImpl=org.apache.ibatis.logging.stdout.StdoutImpl
mybatis-plus.configuration.mapUnderscoreToCamelCase=true
#datasource config
spring.datasource.driverClassName=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql:///books?useUnicode=true&useSSL=false&characterEncoding=UTF-8
spring.datasource.username=root
spring.datasource.password=12345678
```

```
# 开启sql日志
mybatis-plus.configuration.logImpl=org.apache.ibatis.logging.stdout.StdoutImpl
#将带有下划线的表字段映射为驼峰格式的实体类属性
mybatis-plus.configuration.mapUnderscoreToCamelCase=true
```

### 3.2 创建实体类

#### 3.2.1 创建 Comment 类

```
package cn.edu.scnu.entity;

import com.baomidou.mybatisplus.annotation.IdType;
import com.baomidou.mybatisplus.annotation.TableId;
import lombok.Data;
import java.io.Serializable;

@Data
public class Comment implements Serializable {

    private static final long serialVersionUID = -1L;

    @TableId(type=IdType.AUTO)
    private int id;

    private String content;

    private String cAuthor;

    private int bookId;
```

```
}
```

### 3.2.2 创建 Books 类

```
package cn.edu.scnu.entity;
import com.baomidou.mybatisplus.annotation.IdType;
import com.baomidou.mybatisplus.annotation.TableId;
import com.baomidou.mybatisplus.annotation.TableField;
import com.baomidou.mybatisplus.annotation.TableName;
import lombok.Data;
import java.io.Serializable;
import java.util.List;
@Data
@TableName(value="tb_book")
//数据库tb_book表和 JavaBean(Books类) 进行映射
public class Books implements Serializable {
    private static final long serialVersionUID = -5644799954031156649L;
    //value与数据库主键列名一致, 若实体类属性名与表主键列名一致可省略value
    @TableId(value="book_id",type= IdType.AUTO)//指定自增策略
    private Integer id;
    private String title;
    private String author;
    private String press;
    private Double price;
    @TableField(exist = false)
    private List<Comment> commentList;
}
```

## 3.3 创建 Mapper 接口

### 3.3.1 创建 BookMapper 接口

```
package cn.edu.scnu.mapper;
import cn.edu.scnu.entity.Books;
import com.baomidou.mybatisplus.core.mapper.BaseMapper;
import org.apache.ibatis.annotations.Mapper;
3 usages
@Mapper
public interface BookMapper extends BaseMapper<Books> {
}
```

### 3.3.2 创建 CommentMapper 接口

```
1 package cn.edu.scnu.mapper;
2 import cn.edu.scnu.entity.Comment;
3 import com.baomidou.mybatisplus.core.mapper.BaseMapper;
4 import org.apache.ibatis.annotations.Mapper;
5 @Mapper
6 public interface CommentMapper extends BaseMapper<Comment> {
7 }
8
```

## 3.4 创建 Service 层类

创建 BookService 类

```
1 package cn.edu.scnu.service;
2 import cn.edu.scnu.entity.Books;
3 import cn.edu.scnu.mapper.BookMapper;
4 import cn.edu.scnu.mapper.CommentMapper;
5 import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;
6 import org.springframework.stereotype.Service;
7 @Service
8 public class BookService extends ServiceImpl<BookMapper, Books> {
9     private BookMapper bookMapper;
10    private CommentMapper commentMapper;
11    public BookService(BookMapper bookMapper, CommentMapper commentMapper) {
12        this.bookMapper = bookMapper;
13        this.commentMapper = commentMapper;
14    }
15 }
```

## 3.5 控制器层

### 3.5.1 创建控制器 BookController

创建 controller 包，在该包中创建 BookController 类。创建构造方法和 index 方法

```
package cn.edu.scnu.controller;
import cn.edu.scnu.service.BookService;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
@Controller
public class BookController {
    private BookService bookService;
```

```

public BookController(BookService bookService) {
    this.bookService = bookService;
}

@RequestMapping("/index")
public String index(){
    return "index";
}
}

```

### 3.5.2 图书的 CRUD 操作

在 BookController 类中创建 allBooks 方法、insertBook 方法、updateBook 方法和 deleteBook 方法。

```

@RequestMapping("/allBooks")
@ResponseBody
public List<Books> allBooks(){
    return bookService.list();
}

@RequestMapping("/insertBook")
@ResponseBody
public String insertBook(Books book){
    bookService.save(book);
    return "添加成功";
}

@RequestMapping("/updateBook")
@ResponseBody
public String updateBook(Books book){
    bookService.updateById(book);
    return "修改成功";
}

@RequestMapping("/deleteBook")
@ResponseBody
public String deleteBook(Integer id){
    bookService.removeById(id);
    return "删除成功";
}

```

## 4. 查询操作

## 4.1 一对多查询

### 4.1.1 控制器层方法

```
@RequestMapping("/showComment")
@ResponseBody
public Books showComment(Integer id){
    return bookService.getById(id);
}
```

### 4.1.2 Service 层方法

```
public Books showComment(Integer id) {
    QueryWrapper<Comment> commentQueryWrapper=new QueryWrapper<>();
    commentQueryWrapper.eq( column: "book_id",id);
    List<Comment> commentList=commentMapper.selectList(commentQueryWrapper);
    Books book=bookMapper.selectById(id);
    book.setCommentList(commentList);
    return book;
}
```

## 4.2 多条件查询

### 4.2.1 控制器层方法

```
@RequestMapping("/searchBook")
@ResponseBody
public List<Books> searchBook(Books book,Double minPrice,Double maxPrice){
    Map<String,Object> map = new HashMap<>();
    map.put("title",book.getTitle());
    map.put("author",book.getAuthor());
    map.put("press",book.getPress());
    if(minPrice==null){
        minPrice=0.0;
    }
    map.put("minPrice",minPrice);
    if( maxPrice==null){
        maxPrice=Double.MAX_VALUE;
    }
    if(maxPrice<minPrice){
        minPrice=0.0;
    }
    map.put("maxPrice",maxPrice);
    List<Books> booksList=bookService.searchBook(map);
    booksList.forEach(System.out::println);
    return booksList;
}
```



## 4.2.2 service 层方法

BookService 类中的方法：

```
public List<Books> searchBook(Map<String, Object> map) {  
    QueryWrapper<Books> bookQueryWrapper=new QueryWrapper<>();  
    bookQueryWrapper.like( column: "title",map.get("title"));  
    bookQueryWrapper.like( column: "author",map.get("author"));  
    bookQueryWrapper.like( column: "press",map.get("press"));  
    bookQueryWrapper.between( column: "price",map.get("minPrice"),map.get("maxPrice"));  
    return bookMapper.selectList(bookQueryWrapper);  
}
```

## 4.2.3 运行结果

← → ↺ ⓘ localhost:8090/index

添加图书

书名:

作者:

出版社:

价格:  -

教材信息							
id	书名	作者	出版社	价格	修改	删除	详情
1	Hadoop+Spark大数据技术 (微课版)	刘彬斌	清华大学出版社	33	<a href="#">修改</a>	<a href="#">删除</a>	<a href="#">详情</a> <a href="#">查看评论</a>
2	PHP Web 程序设计与项目案例开发	马石安	清华大学出版社	59.8	<a href="#">修改</a>	<a href="#">删除</a>	<a href="#">详情</a> <a href="#">查看评论</a>
3	Java EE框架整合开发入门到实践Spring+SpringMVC+MyBatis	陈恒	清华大学出版社	69.8	<a href="#">修改</a>	<a href="#">删除</a>	<a href="#">详情</a> <a href="#">查看评论</a>

## 5. 排序

### 5.1 查看程序入口

```
function sortBook(btn){  
    var by=$(btn).attr('id').substring(2);  
    $.post("/sortBook",{by:by},function(result){  
        $("#tab").html("<caption><b>教材信息</b></caption>\n" +  
            "\t<tr style='border-width:1px;border-style:dotted;'>\n" +  
            "\t\t<th><a href='javascript:void(0)' onclick='sortBook(this)' id='s_id'>id</a></th>" +  
            "<th><a href='javascript:void(0)' onclick='sortBook(this)' id='s_title'>书名</a></th>" +  
            "<th><a href='javascript:void(0)' onclick='sortBook(this)' id='s_author'>作者</a></th>" +  
            "<th><a href='javascript:void(0)' onclick='sortBook(this)' id='s_press'>出版社</a></th>" +  
            "<th><a href='javascript:void(0)' onclick='sortBook(this)' id='s_price'>价格</a></th>" +  
            "<th>修改</th><th>删除</th><th>详情</th>\n" +  
            "\t</tr>");  
        $.each(result,function(key,value){  
            var id= value.id;  

```

### 5.2 控制器层方法

```
@RequestMapping("/sortBook")  
@ResponseBody  
public List<Books> sortBook(String by){  
    return bookService.sortBook(by);  
}
```

### 5.3 service 层方法

```

public List<Books> sortBook(String by) {
    QueryWrapper<Books> bookQueryWrapper=new QueryWrapper<>();
    bookQueryWrapper.orderByAsc(by);
    return bookMapper.selectList(bookQueryWrapper);
}

```

## 5.4 运行结果

按价格排序的结果：

← → ↺ ⓘ localhost:8090/index

[添加图书](#)

书名:   
 作者:   
 出版社:   
 价格:  -

教材信息

id	书名	作者	出版社	价格	修改	删除	详情
5	SpringCloud微服务架构开发	黑马程序员	人民邮电出版社	43	<a href="#">修改</a>	<a href="#">删除</a>	<a href="#">详情</a> <a href="#">查看评论</a>
4	SpringBoot企业级开发教程	黑马程序员	人民邮电出版社	56	<a href="#">修改</a>	<a href="#">删除</a>	<a href="#">详情</a> <a href="#">查看评论</a>
2	PHPWeb 程序设计与项目案例开发	马石安	清华大学出版社	59.8	<a href="#">修改</a>	<a href="#">删除</a>	<a href="#">详情</a> <a href="#">查看评论</a>
1	Hadoop+Spark大数据技术 (微课版)	刘彬斌	清华大学出版社	69	<a href="#">修改</a>	<a href="#">删除</a>	<a href="#">详情</a> <a href="#">查看评论</a>
3	JavaEE框架整合开发入门到实践Spring+SpringMVC+MyBatis	陈恒	清华大学出版社	69.8	<a href="#">修改</a>	<a href="#">删除</a>	<a href="#">详情</a> <a href="#">查看评论</a>
6	Java自学宝典	黑马程序员	清华大学出版社	73	<a href="#">修改</a>	<a href="#">删除</a>	<a href="#">详情</a> <a href="#">查看评论</a>