



链滴

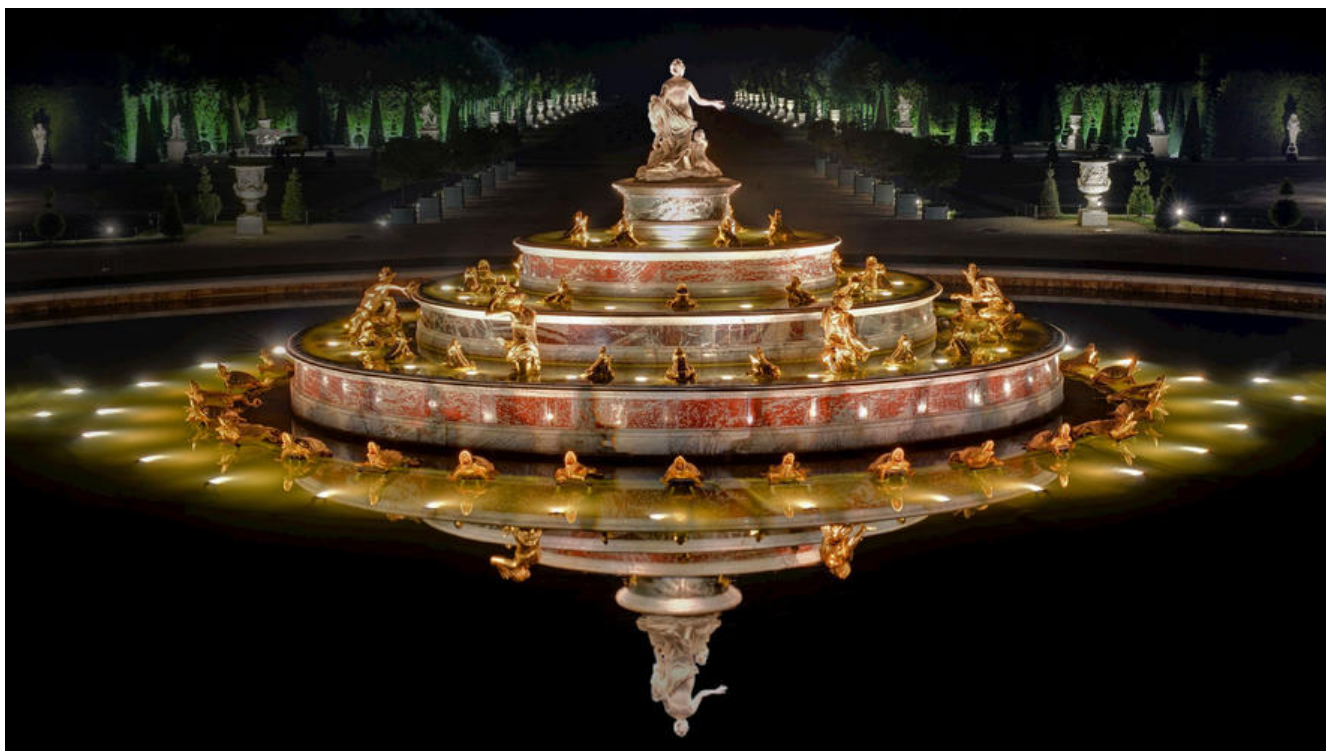
# SpringMVC 集成 MyBatis-Plus 3.x 版本

作者: [kangaroo1122](#)

原文链接: <https://ld246.com/article/1575092987567>

来源网站: [链滴](#)

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



对于集成新版本的MyBatis-Plus, Spring Boot版本官方提供了详细的集成方案, 但是Spring MVC本显得相对来说就有点儿简单、不完善, 毕竟官方推荐使用Spring Boot方式集成。研究了一下, 做记录。

## 1 介绍

**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, 简化开发、提高效率而生。具体的介绍不用多说, 直接看官方文档介绍就行: <https://mybatis.plus/guide/>。

这里集成版本使用的是MyBatis-Plus目前的最新版本: **3.2.0**版本。

2.x版本集成教程网上有很多, 但是2.x版本的JDK最高只能 **1.7**。

## 2 集成

### 2.1 要求

全新的 **MyBatis-Plus 3.0** 版本基于 **JDK8**, 提供了 **lambda** 形式的调用, 所以安装集成 MP3.0 要求下:

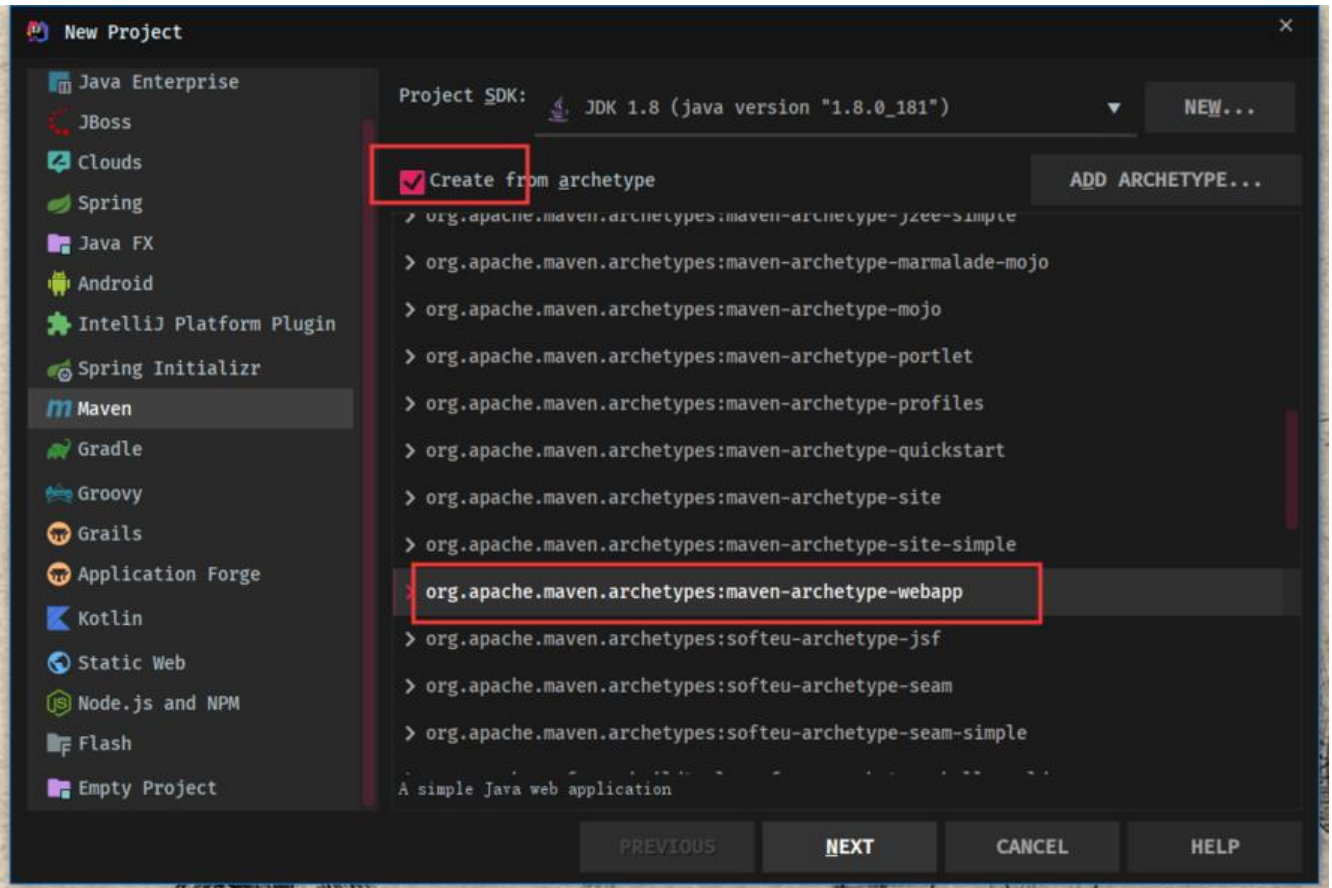
- JDK 8+
- Maven or Gradle

IDE这里选择的是IntelliJ IDEA, 其他也无所谓, 大同小异。

Spring 版本这里使用的是 **5.2.0.RELEASE**

### 2.2 创建maven项目

创建项目比较简单，如图：

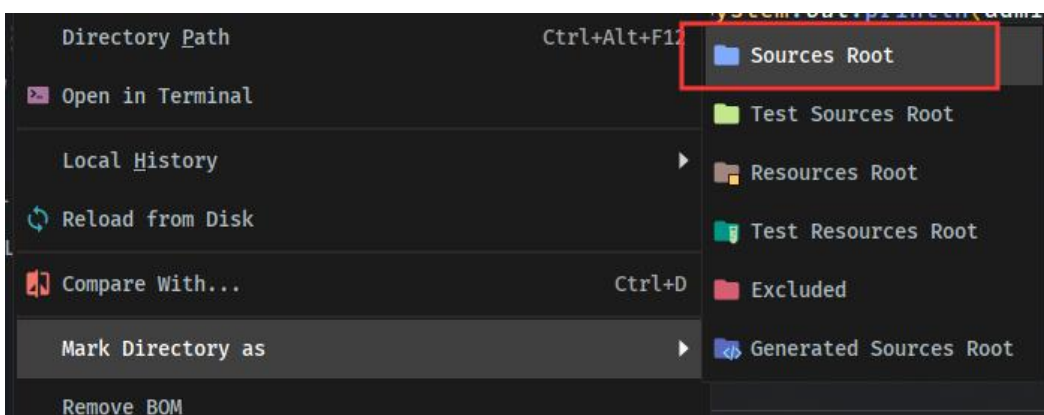


填写相关信息，完成创建。

这里给个创建项目的参考链接：<https://blog.csdn.net/MaNongXf/article/details/83418353>

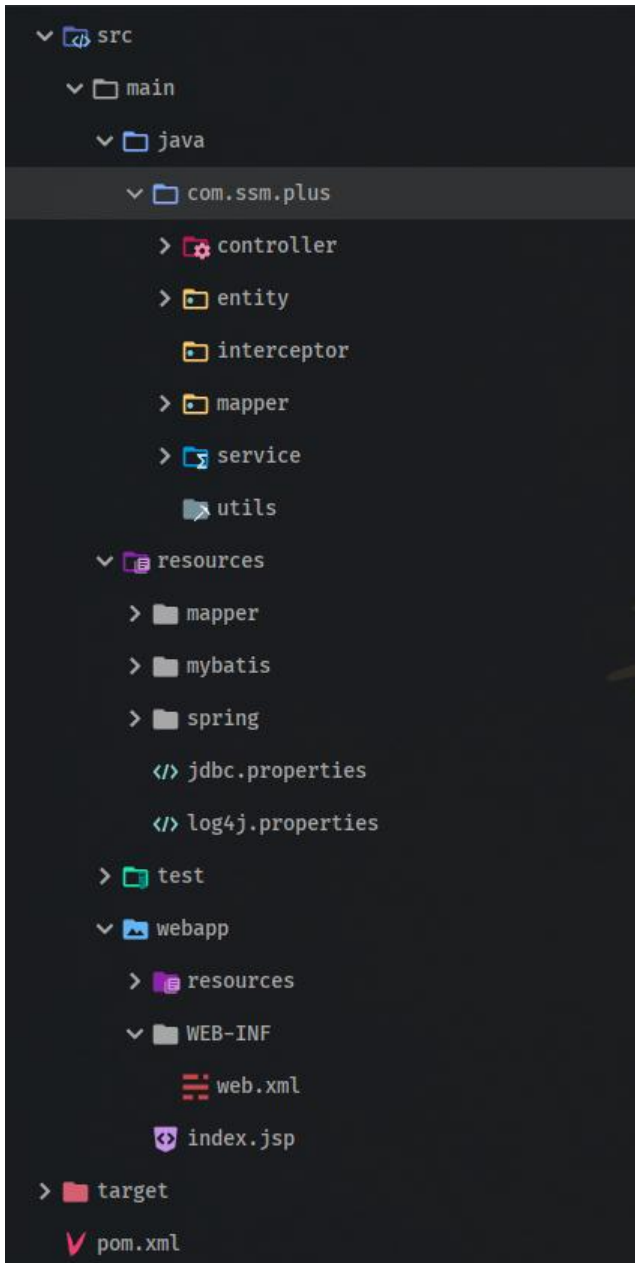
看到输出Hello World页面这一步，项目创建就算是完成了。

在main下新建Directory: java，选中java鼠标右键，选择如下：



同理创建resources，test两个文件夹，分别标记为Resources Root和Test Sources Root

项目结构如下：



其中, java > resources下的mapper文件夹是用于存放自定义的mybatis的xml文件  
mybatis文件夹用于存放mybatis的配置文件

## 2.3 配置文件

### 2.3.1 pom.xml

```
<?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 http://maven.apache.org/xsd  
maven-4.0.0.xsd">  
    <modelVersion>4.0.0</modelVersion>
```

```

<groupId>com.ssm.plus</groupId>
<artifactId>ssm-plus</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>war</packaging>

<name>ssm-plus Maven Webapp</name>
<!-- FIXME change it to the project's website -->
<url>http://www.example.com</url>

<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <maven.compiler.source>1.8</maven.compiler.source>
  <maven.compiler.target>1.8</maven.compiler.target>
  <servlet.version>3.1.0</servlet.version>
  <jsp.version>2.3.1</jsp.version>
  <jstl.version>1.2</jstl.version>
  <mybatisplus.version>3.2.0</mybatisplus.version>
  <spring.version>5.2.0.RELEASE</spring.version>
  <mysql.version>5.1.40</mysql.version>
  <log4j.version>1.2.17</log4j.version>
  <lombok.version>1.18.10</lombok.version>
  <fastjson.version>1.2.62</fastjson.version>
  <dbcp2.version>2.6.0</dbcp2.version>
  <fileupload.version>1.3.3</fileupload.version>
  <io.version>2.6</io.version>
</properties>

<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
  </dependency>
  <!-- 加入servlet的依赖 -->
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>${servlet.version}</version>
    <scope>provided</scope>
  </dependency>
  <!-- 加入jsp的依赖 -->
  <dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>javax.servlet.jsp-api</artifactId>
    <version>${jsp.version}</version>
    <scope>provided</scope>
  </dependency>
  <!-- 加入jstl的依赖 -->
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>${jstl.version}</version>
  </dependency>

```

```
<!-- 加入mybatis-plus的依赖 -->
<dependency>
  <groupId>com.baomidou</groupId>
  <artifactId>mybatis-plus</artifactId>
  <version>${mybatisplus.version}</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>${lombok.version}</version>
  <scope>provided</scope>
</dependency>
<!-- 加入spring的依赖 -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-test</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-aspects</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-aop</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-expression</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-jdbc</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-tx</artifactId>
  <version>${spring.version}</version>
</dependency>
```



```

<!-- 加入springmvc的依赖 -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>${spring.version}</version>
</dependency>
<!-- 加入mysql的依赖 -->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>${mysql.version}</version>
</dependency>
<!-- 加入log4j的依赖 -->
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>${log4j.version}</version>
</dependency>
<!-- 加入dbcp2数据库依赖 -->
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-dbcp2</artifactId>
  <version>${dbcp2.version}</version>
</dependency>
<!-- 加入fileupload上传依赖 -->
<dependency>
  <groupId>commons-fileupload</groupId>
  <artifactId>commons-fileupload</artifactId>
  <version>${fileupload.version}</version>
</dependency>
<!-- 加入io依赖 -->
<dependency>
  <groupId>commons-io</groupId>
  <artifactId>commons-io</artifactId>
  <version>${io.version}</version>
</dependency>
<!-- FastJson -->
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>fastjson</artifactId>
  <version>${fastjson.version}</version>
</dependency>
</dependencies>

<build>
  <finalName>ssm-plus</finalName>
  <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (m
y be moved to parent pom) -->
    <plugins>
      <plugin>
        <artifactId>maven-clean-plugin</artifactId>
        <version>3.1.0</version>
      </plugin>
      <!-- see http://maven.apache.org/ref/current/maven-core/default-bindings.html#Pl

```

```

gin_bindings_for_war_packaging -->
  <plugin>
    <artifactId>maven-resources-plugin</artifactId>
    <version>3.0.2</version>
  </plugin>
  <plugin>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>3.8.0</version>
  </plugin>
  <plugin>
    <artifactId>maven-surefire-plugin</artifactId>
    <version>2.22.1</version>
  </plugin>
  <plugin>
    <artifactId>maven-war-plugin</artifactId>
    <version>3.2.2</version>
  </plugin>
  <plugin>
    <artifactId>maven-install-plugin</artifactId>
    <version>2.5.2</version>
  </plugin>
  <plugin>
    <artifactId>maven-deploy-plugin</artifactId>
    <version>2.8.2</version>
  </plugin>
</plugins>
</pluginManagement>
</build>
</project>

```

在普通SSM项目中，需要引入mybatis和mybatis-spring两个依赖，这里将这两个依赖替换成mybatis plus的依赖，mybatis-plus会自动引入这两个依赖，故这里不用重复引入，以免冲突。

## 2.3.2 mybatis-config.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD SQL Map Config 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-config.dtd">
<!--
| plugins在配置文件中的位置必须符合要求，否则会报错，顺序如下：
| properties?, settings?,
| typeAliases?, typeHandlers?,
| objectFactory?,objectWrapperFactory?,
| plugins?,
| environments?, databaseIdProvider?, mappers?
|-->
<configuration>
  <!--
  | 全局配置设置
  |
  | 可配置选项          默认值    描述
  |
  | aggressiveLazyLoading    true,   当设置为 'true' 的时候，懒加载的对象可能被任何懒属

```



全部加载。否则，每个属性都按需加载。

```
| multipleResultSetsEnabled true, 允许和不允许单条语句返回多个数据集（取决于驱动需求

| useColumnLabel true, 使用列标签代替列名称。不同的驱动器有不同的作法。参考
下驱动器文档，或者用这两个不同的选项进行测试一下。
| useGeneratedKeys false, 允许JDBC生成主键。需要驱动器支持。如果设为了true，
个设置将强制使用被生成的主键，有一些驱动器不兼容不过仍然可以执行。
| autoMappingBehavior PARTIAL, 指定MyBatis是否并且如何来自动映射数据表字段与对
的属性。PARTIAL将只自动映射简单的，没有嵌套的结果。FULL将自动映射所有复杂的结果。
| defaultExecutorType SIMPLE, 配置和设定执行器，SIMPLE执行器执行其它语句。REUSE
执行器可能重复使用prepared statements语句，BATCH执行器可以重复执行语句和批量更新。
| defaultStatementTimeout null, 设置一个时限，以决定让驱动器等待数据库回应的多长时间
为超时
|-->
<settings>
  <!-- 这个配置使全局的映射器启用或禁用缓存 -->
  <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 -->
  <setting name="lazyLoadingEnabled" value="true"/>
  <setting name="multipleResultSetsEnabled" value="true"/>
  <setting name="useColumnLabel" value="true"/>
  <setting name="defaultExecutorType" value="REUSE"/>
  <setting name="defaultStatementTimeout" value="25000"/>
</settings>

</configuration>
```

### 2.3.3 spring.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:context="http://www.springframework.org/schema/context"
  xmlns:util="http://www.springframework.org/schema/util"
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springf
amework.org/schema/beans/spring-beans.xsd http://www.springframework.org/schema/con
ext https://www.springframework.org/schema/context/spring-context.xsd http://www.springf
amework.org/schema/util https://www.springframework.org/schema/util/spring-util.xsd" >

  <!-- 引入属性文件 -->
  <context:property-placeholder location="classpath:jdbc.properties"/>
  <util:properties id="configProperties" location="classpath:jdbc.properties" />

  <!-- Service包(自动注入) -->
  <context:component-scan base-package="com.ssm.plus.service"/>

  <import resource="classpath:spring/spring-mybatis.xml"/>
</beans>
```

### 2.3.4 spring-mvc.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:mvc="http://www.springframework.org/schema/mvc"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd http://www.springframework.org/schema/mvc https://www.springframework.org/schema/mvc/spring-mvc.xsd http://www.springframework.org/schema/context https://www.springframework.org/schema/context/spring-context.xsd" >

    <mvc:default-servlet-handler/>

    <!-- Controller包(自动注入) -->
    <context:component-scan base-package="com.ssm.plus.controller"/>

    <!-- 注解映射器, 适配器 -->
    <mvc:annotation-driven >
        <mvc:message-converters register-defaults="true">
            <!-- 避免IE执行AJAX时,返回JSON出现下载文件 -->
            <!-- FastJson -->
            <bean id="fastJsonHttpMessageConverter"
                class="com.alibaba.fastjson.support.spring.FastJsonHttpMessageConverter">
                <property name="supportedMediaTypes">
                    <list>
                        <!-- 这里顺序不能反, 一定先写text/html,不然ie下出现下载提示 -->
                        <value>text/html;charset=UTF-8</value>
                        <value>application/json;charset=UTF-8</value>
                    </list>
                </property>
                <property name="features">
                    <array value-type="com.alibaba.fastjson.serializer.SerializerFeature">
                        <!-- 避免循环引用 -->
                        <value>DisableCircularReferenceDetect</value>
                        <!-- 是否输出值为null的字段 -->
                        <value>WriteMapNullValue</value>
                        <!-- 数值字段如果为null,输出为0,而非null -->
                        <value>WriteNullNumberAsZero</value>
                        <!-- 字符类型字段如果为null,输出为",而非null -->
                        <value>WriteNullStringAsEmpty</value>
                        <!-- List字段如果为null,输出为[],而非null -->
                        <value>WriteNullListAsEmpty</value>
                        <!-- Boolean字段如果为null,输出为false,而非null -->
                        <value>WriteNullBooleanAsFalse</value>
                    </array>
                </property>
            </bean>
        </mvc:message-converters>
    </mvc:annotation-driven>

    <!-- 资源映射 -->
    <mvc:resources location="/resources/" mapping="/resources/**" />

    <!-- 视图解析器 -->
    <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="prefix" value="/WEB-INF/jsp/" />
        <property name="suffix" value=".jsp" />
    </bean>

```

```

</bean>

<!-- 定义文件上传解析器 -->
<bean id="multipartResolver"
    class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
    <!-- 设定默认编码 -->
    <property name="defaultEncoding" value="UTF-8"/>
    <!-- 设定文件上传的最大值5MB, 5*1024*1024 -->
    <property name="maxUploadSize" value="5242880"/>
</bean>

<!-- 拦截器 -->
<!--<mvc:interceptors>
    &lt;!&ndash; 多个拦截器,顺序执行
    用户身份校验的拦截器 &ndash;&gt;
    <mvc:interceptor>
        <mvc:mapping path="/admin/**" />
        <bean class="com.ssm.plus.interceptor.LoginInterceptor"> </bean>
    </mvc:interceptor>

</mvc:interceptors>-->
</beans>

```

## 2.3.5 spring-mybatis.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:tx="http://www.springf
amework.org/schema/tx"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springf
amework.org/schema/beans/spring-beans.xsd http://www.springframework.org/schema/tx ht
tp://www.springframework.org/schema/tx/spring-tx.xsd http://www.springframework.org/sch
ma/aop https://www.springframework.org/schema/aop/spring-aop.xsd">

    <!-- 数据库连接池 -->
    <bean id="dataSource" class="org.apache.commons.dbcp2.BasicDataSource" destroy-met
od="close">
        <property name="driverClassName" value="{jdbc.driver}" />
        <property name="url" value="{jdbc.url}" />
        <property name="username" value="{jdbc.username}" />
        <property name="password" value="{jdbc.password}" />
        <property name="maxTotal" value="10" />
        <property name="minIdle" value="5" />
        <property name="initialSize" value="5" />
    </bean>

    <!-- Spring整合Mybatis, 更多查看文档: http://mp.baomidou.com -->
    <bean id="sqlSessionFactory" class="com.baomidou.mybatisplus.extension.spring.Mybatis
sqlSessionFactoryBean">
        <property name="dataSource" ref="dataSource"/>
        <!-- 自动扫描Mapping.xml文件 -->
        <property name="mapperLocations" value="classpath:mapper/*.xml"/>
        <property name="configLocation" value="classpath:mybatis/mybatis-config.xml"/>

```

```

<property name="typeAliasesPackage" value="com.ssm.plus.entity.*"/>
<!-- 分页 -->
<property name="plugins">
  <array>
    <bean class="com.baomidou.mybatisplus.extension.plugins.PaginationInterceptor"
  >
    </array>
  </property>
<!-- 全局配置注入 -->
<property name="globalConfig" ref="globalConfig" />
</bean>
<bean id="globalConfig" class="com.baomidou.mybatisplus.core.config.GlobalConfig">
  <property name="dbConfig" ref="dbConfig"/>
</bean>
<bean id="dbConfig" class="com.baomidou.mybatisplus.core.config.GlobalConfig.DbConfi
">
  <property name="keyGenerator" ref="keyGenerator"/>
</bean>
<bean id="keyGenerator" class="com.baomidou.mybatisplus.extension.incrementer.DB2Ke
Generator"/>

<!-- MyBatis 动态扫描 -->
<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
  <property name="basePackage" value="com.ssm.plus.mapper"/>
</bean>

<!-- 配置事务管理 -->
<bean name="transactionManager" class="org.springframework.jdbc.datasource.DataSour
eTransactionManager">
  <property name="dataSource" ref="dataSource"/>
</bean>

<!-- 事务管理 属性 -->
<tx:advice id="transactionAdvice" transaction-manager="transactionManager">
  <tx:attributes>
    <tx:method name="add*" propagation="REQUIRED"/>
    <tx:method name="append*" propagation="REQUIRED"/>
    <tx:method name="save*" propagation="REQUIRED"/>
    <tx:method name="update*" propagation="REQUIRED"/>
    <tx:method name="modify*" propagation="REQUIRED"/>
    <tx:method name="edit*" propagation="REQUIRED"/>
    <tx:method name="insert*" propagation="REQUIRED"/>
    <tx:method name="delete*" propagation="REQUIRED"/>
    <tx:method name="remove*" propagation="REQUIRED"/>
    <tx:method name="repair" propagation="REQUIRED"/>

    <tx:method name="get*" propagation="REQUIRED" read-only="true"/>
    <tx:method name="find*" propagation="REQUIRED" read-only="true"/>
    <tx:method name="load*" propagation="REQUIRED" read-only="true"/>
    <tx:method name="search*" propagation="REQUIRED" read-only="true"/>
    <tx:method name="datagrid*" propagation="REQUIRED" read-only="true"/>

    <tx:method name="*" propagation="REQUIRED" read-only="true"/>
  </tx:attributes>

```

```

</tx:advice>

<!-- 配置切面 -->
<aop:config>
  <aop:pointcut id="transactionPointcut" expression="execution(* com.ssm.plus.service..*.
(..))"/>
  <aop:advisor pointcut-ref="transactionPointcut" advice-ref="transactionAdvice"/>
</aop:config>
</beans>

```

## 2.3.6 web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.
3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/jav
ee/web-app_3_0.xsd">

  <display-name>ssm-plus</display-name>

  <!-- 加载Spring配置文件 -->
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:spring/spring.xml</param-value>
  </context-param>

  <!-- 字符集 过滤器 -->
  <filter>
    <filter-name>encodingFilter</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>encodingFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>

  <!-- Spring监听器 -->
  <listener>
    <description>Spring监听器</description>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
  </listener>
  <!-- 防止Spring内存溢出监听器 -->
  <listener>
    <listener-class>org.springframework.web.util.IntrospectorCleanupListener</listener-class>
  </listener>

  <!-- Spring MVC -->
  <servlet>
    <servlet-name>SpringMVC</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

```

```

<init-param>
  <description>SpringMVC</description>
  <param-name>contextConfigLocation</param-name>
  <param-value>classpath:spring/spring-mvc.xml</param-value>
</init-param>
<load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>SpringMVC</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping>

<!-- Session超时时间 -->
<session-config>
  <session-timeout>15</session-timeout>
</session-config>
</web-app>

```

### 2.3.7 jdbc.properties

```

jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost:3306/db_ssmp?useUnicode=true&characterEncoding=utf-8&
seSSL=false
jdbc.username=
jdbc.password=

```

### 2.3.8 log4j.properties

```

# Global logging configuration
log4j.rootLogger=DEBUG, stdout
# MyBatis logging configuration...
log4j.logger.org.mybatis.example.BlogMapper=TRACE
# Console output...
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%5p [%t] - %m%n

```

## 2.4 测试

### 2.4.1 实体类Admin.java

```

package com.ssm.plus.entity;
import com.baomidou.mybatisplus.annotation.TableName;
import lombok.Data;

/**
 * Created by kangaroo on 2019/11/18 21:16.
 * QQ: 326170945
 * Description:
 */
@Data
@TableName("t_admin")

```

```
public class Admin {
    private long id;
    private String name;
    private String username;
    private String password;
}
```

- **注:**

- @Data 注解是lombok提供的，需要开发工具安装lombok的插件才能使用，如果没有安装，就删这个注解，手动生成getter和setter方法，并在pom.xml中删除lombok的依赖。

- @TableName 注解是mybatis-plus提供的，用于指定数据库表名，默认实体类名。具体注解，可参考官方文档。

## 2.4.2 dao层接口AdminMapper.java

```
package com.ssm.plus.mapper;
import com.baomidou.mybatisplus.core.mapper.BaseMapper;
import com.ssm.plus.entity.Admin;

/**
 * Created by kangaroo on 2019/11/18 21:15.
 * QQ: 326170945
 * Description:
 */
public interface AdminMapper extends BaseMapper<Admin> {
}
```

## 2.4.3 测试方法类MpTest.java

```
package com.ssm.plus.test;

import com.ssm.plus.entity.Admin;
import com.ssm.plus.mapper.AdminMapper;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

import java.util.List;

/**
 * Created by kangaroo on 2019/11/18 19:52.
 * QQ: 326170945
 * Description:
 */
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = {"classpath*:spring/spring.xml"})
public class MpTest {

    @Autowired
    private AdminMapper adminMapper;
```



```
@Test
public void select(){
    System.out.println("查询测试: ");
    List<Admin> adminList = adminMapper.selectList(null);
    System.out.println(adminList.size());
    adminList.forEach(System.out::println);
}
}
```

测试结果:



A terminal window with a black background and white text. At the top, it shows the version '3.2.0'. Below that, it prints '查询测试:' followed by the number '2'. Then, it lists two Admin objects: 'Admin(id=1, name=admin, username=1000011@qq.com, password=1000011)' and 'Admin(id=2, name=超级管理员, username=1008611@qq.com, password=1008611)'.

```
3.2.0
查询测试:
2
Admin(id=1, name=admin, username=1000011@qq.com, password=1000011)
Admin(id=2, name=超级管理员, username=1008611@qq.com, password=1008611)
```

### 3 Demo下载

demo下载地址: <https://github.com/kangaroo1122/ssm-plus>