SSM框架整合

Spring + Spring MVC + MyBatis

Spring MVC 负责实现 MVC 设计模式，MyBatis 负责数据持久层，Spring 负责管理 Spring MVC 和 MyBatis 相关对象的创建和依赖注入。

- 创建 Maven 工程，pom.xml

```xml
<dependencies>
  <!-- SpringMVC -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>5.0.11.RELEASE</version>
  </dependency>

  <!-- Spring JDBC -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>5.0.11.RELEASE</version>
  </dependency>

  <!-- Spring AOP -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-aop</artifactId>
    <version>5.0.11.RELEASE</version>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-aspects</artifactId>
    <version>5.0.11.RELEASE</version>
  </dependency>

  <!-- MyBatis -->
  <dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
    <version>3.4.5</version>
  </dependency>

  <!-- MyBatis整合Spring -->
  <dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis-spring</artifactId>
    <version>1.3.1</version>
  </dependency>

  <!-- MySQL驱动 -->
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.11</version>
  </dependency>

  <!-- C3P0 -->
  <dependency>
    <groupId>c3p0</groupId>
    <artifactId>c3p0</artifactId>
    <version>0.9.1.2</version>
  </dependency>

  <!-- JSTL -->
  <dependency>
    <groupId>jstl</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
  </dependency>

  <!-- ServletAPI -->
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>3.1.0</version>
  </dependency>

  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.6</version>
    <scope>provided</scope>
  </dependency>
</dependencies>
```

- web.xml 中配置 SpringMVC、Spring、字符编码过滤器、加载静态资源。

```
<!DOCTYPE web-app PUBLIC
 "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
 "http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>
  <display-name>Archetype Created Web Application</display-name>
  <!-- 启动Spring -->
  <context-param>
```

```xml
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:spring.xml</param-value>
  </context-param>
  <listener>
    <listener-
class>org.springframework.web.context.ContextLoaderListener</listener-class>
  </listener>

  <!-- Spring MVC -->
  <servlet>
    <servlet-name>dispatcherServlet</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-
class>
    <init-param>
      <param-name>contextConfigLocation</param-name>
      <param-value>classpath:springmvc.xml</param-value>
    </init-param>
  </servlet>

  <servlet-mapping>
    <servlet-name>dispatcherServlet</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>

  <!-- 字符编码过滤器 -->
  <filter>
    <filter-name>characterEncodingFilter</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>characterEncodingFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>

  <!-- 加载静态资源 -->
  <servlet-mapping>
    <servlet-name>default</servlet-name>
    <url-pattern>*.js</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>default</servlet-name>
    <url-pattern>*.css</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
```

```
    <servlet-name>default</servlet-name>
    <url-pattern>*.jpg</url-pattern>
  </servlet-mapping>
</web-app>
```

- 在 spring.xml 中配置 MyBatis 和 Spring 的整合。

```
<?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:p="http://www.springframework.org/schema/p"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.2.xsd
  http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.3.xsd
">

    <!-- 整合MyBatis -->
    <bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource">
        <property name="user" value="root"></property>
        <property name="password" value="root"></property>
        <property name="jdbcUrl" value="jdbc:mysql://localhost:3306/test?
useUnicode=true&amp;characterEncoding=UTF-8"></property>
        <property name="driverClass" value="com.mysql.cj.jdbc.Driver">
</property>
        <property name="initialPoolSize" value="5"></property>
        <property name="maxPoolSize" value="10"></property>
    </bean>

    <!-- 配置MyBatis SqlSessionFactory -->
    <bean id="sqlSessionFactory"
class="org.mybatis.spring.SqlSessionFactoryBean">
        <property name="dataSource" ref="dataSource"></property>
        <property name="mapperLocations"
value="classpath:com/southwind/repository/*.xml"></property>
        <property name="configLocation" value="classpath:config.xml">
</property>
    </bean>

    <!-- 扫描自定义的Mapper接口 -->
    <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
        <property name="basePackage" value="com.southwind.repository">
</property>
    </bean>

</beans>
```

- config.xml 配置一些 MyBatis 辅助信息，比如打印 SQL 等。

```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>
    <settings>
        <!-- 打印SQL-->
        <setting name="logImpl" value="STDOUT_LOGGING" />
    </settings>

    <typeAliases>
        <!-- 指定一个包名，MyBatis会在包名下搜索需要的JavaBean-->
        <package name="com.southwind.entity"/>
    </typeAliases>

</configuration>
```

- 配置 springmvc.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:context="http://www.springframework.org/schema/context"
       xmlns:mvc="http://www.springframework.org/schema/mvc"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
        http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.0.xsd
        http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-3.2.xsd">

    <!-- 启动注解驱动 -->
    <mvc:annotation-driven></mvc:annotation-driven>

    <!-- 扫描业务代码 -->
    <context:component-scan base-package="com.southwind"></context:component-
scan>

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

</beans>
```

- 实体类

```java
package com.southwind.entity;

import lombok.Data;

@Data
public class User {
    private long id;
    private String name;
    private String password;
    private double score;
}
```

- UserRepository

```java
package com.southwind.repository;

import com.southwind.entity.User;

import java.util.List;

public interface UserRepository {
    public List<User> findAll();
}
```

- UserRepository.xml

```xml
<?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.southwind.repository.UserRepository">
    <select id="findAll" resultType="User">
        select * from user
    </select>
</mapper>
```

- UserService

```java
package com.southwind.service;

import com.southwind.entity.User;

import java.util.List;

public interface UserService {
    public List<User> findAll();
}
```

- UserServiceImpl

```java
package com.southwind.service.impl;

import com.southwind.entity.User;
import com.southwind.repository.UserRepository;
import com.southwind.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class UserServiceImpl implements UserService {

    @Autowired
    private UserRepository userRepository;

    @Override
    public List<User> findAll() {
        return userRepository.findAll();
    }
}
```

- UserHandler

```java
package com.southwind.controller;

import com.southwind.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

@Controller
@RequestMapping("/user")
public class UserHandler {
    @Autowired
    private UserService userService;

    @GetMapping("/findAll")
    public ModelAndView findAll(){
        ModelAndView modelAndView = new ModelAndView();
        modelAndView.setViewName("index");
        modelAndView.addObject("list",userService.findAll());
        return modelAndView;
    }
}
```