

# 08-SSM框架整合

## 1、三大框架整合

在老期的项目中，一般都是使用ssm项目做开发的，虽然现在的主流开发是springboot来做开发，但是ssm的基本整合还是需要掌握的。

**整合SSM框架要做哪些事情：**

- SpringMVC: pom
  - 1. web.xml
    - a. 前端调度器servlet
    - b. 编码过滤器filter
    - c. 支持rest的过滤器
  - 2. springmvc.xml
    - a. 扫描controller包
    - b. 添加<annotation-driver>
    - c. 视图解析器
    - d. 静态资源解析
  - 3. 添加控制器类...
- Spring:
  - 1. web.xml
    - a. 监听器（在启动web容器时加载）
  - 2. spring.xml
    - a. 扫描所有除了controller包的其他包
    - b. 声明式事务
- MyBatis
  - 1. 需要和spring整合
    - a. 将sqlSessionFactory 配置为spring的bean
      - i. 数据源 配置为spring的bean

## ii. 全局配置文件

## iii. 所有mapper映射文件

### b. 将mapper接口的包交给spring

### 2. 加入全局配置文件

## 1. 导入pom文件

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6
7   <groupId>cn.tulingxueyuan</groupId>
8   <artifactId>ssm</artifactId>
9   <version>1.0-SNAPSHOT</version>
10
11   <properties>
12     <SPRING.VERSION>5.2.6.RELEASE</SPRING.VERSION>
13     <ASPECTJWEAVER.VERSION>1.9.5</ASPECTJWEAVER.VERSION>
14   </properties>
15
16   <dependencies>
17     <!--springmvc-->
18     <dependency>
19       <groupId>org.springframework</groupId>
20       <artifactId>spring-webmvc</artifactId>
21       <version>${SPRING.VERSION}</version>
22     </dependency>
23     <!--spring-->
24     <dependency>
25       <groupId>org.springframework</groupId>
26       <artifactId>spring-aop</artifactId>
27       <version>${SPRING.VERSION}</version>
28     </dependency>
29     <dependency>
30       <groupId>org.aspectj</groupId>
31       <artifactId>aspectjweaver</artifactId>
```

```
32 <version>${ASPECTJWEAVER.VERSION}</version>
33 </dependency>
34 <dependency>
35 <groupId>org.springframework</groupId>
36 <artifactId>spring-aspects</artifactId>
37 <version>${SPRING.VERSION}</version>
38 </dependency>
39 <dependency>
40 <groupId>org.springframework</groupId>
41 <artifactId>spring-orm</artifactId>
42 <version>${SPRING.VERSION}</version>
43 </dependency>
44
45 <!--spring-mybatis 适配器-->
46 <dependency>
47 <groupId>org.mybatis</groupId>
48 <artifactId>mybatis-spring</artifactId>
49 <version>2.0.5</version>
50 </dependency>
51
52
53 <!--mybatis核心依赖-->
54 <dependency>
55 <groupId>org.mybatis</groupId>
56 <artifactId>mybatis</artifactId>
57 <version>3.5.5</version>
58 </dependency>
59 <!--连接池-->
60 <dependency>
61 <groupId>com.alibaba</groupId>
62 <artifactId>druid</artifactId>
63 <version>1.1.21</version>
64 </dependency>
65 <!--mysql 对应版本的连接器驱动-->
66 <dependency>
67 <groupId>mysql</groupId>
68 <artifactId>mysql-connector-java</artifactId>
69 <version>5.1.47</version>
70 </dependency>
71 <!-- log start -->
```

```
72 <dependency>
73 <groupId>org.slf4j</groupId>
74 <artifactId>slf4j-api</artifactId>
75 <version>1.7.30</version>
76 </dependency>
77
78
79 <dependency>
80 <groupId>ch.qos.logback</groupId>
81 <artifactId>logback-classic</artifactId>
82 <version>1.2.3</version>
83 </dependency>
84 <!-- log end -->
85
86
87 <!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jack
son-core -->
88 <dependency>
89 <groupId>com.fasterxml.jackson.core</groupId>
90 <artifactId>jackson-core</artifactId>
91 <version>2.10.3</version>
92 </dependency>
93 <!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jack
son-databind -->
94 <dependency>
95 <groupId>com.fasterxml.jackson.core</groupId>
96 <artifactId>jackson-databind</artifactId>
97 <version>2.10.3</version>
98 </dependency>
99 <!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jack
son-annotations -->
100 <dependency>
101 <groupId>com.fasterxml.jackson.core</groupId>
102 <artifactId>jackson-annotations</artifactId>
103 <version>2.10.3</version>
104 </dependency>
105
106 <dependency>
107 <groupId>junit</groupId>
108 <artifactId>junit</artifactId>
109 <version>4.12</version>
```

```
110 <scope>test</scope>
111 </dependency>
112 </dependencies>
113
114 </project>
```

## 2. 编写各个框架的配置文件

### web.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
5   version="3.1">
6
7
8   <!--spring 基于web应用的启动-->
9   <listener>
10     <listener-
11       class>org.springframework.web.context.ContextLoaderListener</listener-
12       class>
13   </listener>
14   <!--全局参数: spring配置文件-->
15   <context-param>
16     <param-name>contextConfigLocation</param-name>
17     <param-value>classpath:spring-core.xml</param-value>
18   </context-param>
19
20   <!--前端调度器servlet-->
21   <servlet>
22     <servlet-name>dispatcherServlet</servlet-name>
23     <servlet-class>org.springframework.web.servlet.DispatcherServlet</servl
24       et-class>
25     <!--设置配置文件的路径-->
26     <init-param>
27       <param-name>contextConfigLocation</param-name>
28       <param-value>classpath:spring-mvc.xml</param-value>
29     </init-param>
30     <!--设置启动即加载-->
31     <load-on-startup>1</load-on-startup>
```

```
31 </servlet>
32 <servlet-mapping>
33 <servlet-name>dispatcherServlet</servlet-name>
34 <url-pattern>/</url-pattern>
35 </servlet-mapping>
36
37 <!--编码过滤器filter-->
38 <filter>
39 <filter-name>characterEncoding</filter-name>
40 <filter-class>org.springframework.web.filter.CharacterEncodingFilter</f
41 <!--设置编码格式-->
42 <init-param>
43 <param-name>encoding</param-name>
44 <param-value>UTF-8</param-value>
45 </init-param>
46 <!-->
47 <init-param>
48 <param-name>forceEncoding</param-name>
49 <param-value>true</param-value>
50 </init-param>
51 </filter>
52 <filter-mapping>
53 <filter-name>characterEncoding</filter-name>
54 <servlet-name>dispatcherServlet</servlet-name>
55 </filter-mapping>
56
57 <!--支持rest的过滤器-->
58 <filter>
59 <filter-name>hiddenHttpMethod</filter-name>
60 <filter-class>org.springframework.web.filter.HiddenHttpMethodFilter</fi
61 </filter>
62 <filter-mapping>
63 <filter-name>hiddenHttpMethod</filter-name>
64 <servlet-name>dispatcherServlet</servlet-name>
65 </filter-mapping>
66
67 </web-app>
```

springmvc.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xmlns:context="http://www.springframework.org/schema/context"
5   xmlns:mvc="http://www.springframework.org/schema/mvc"
6   xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd http://www.springframework.org/schema/context https://www.springframework.org/schema/context/spring-context.xsd http://www.springframework.org/schema/mvc https://www.springframework.org/schema/mvc/spring-mvc.xsd">
7
8   <!--springmvc只扫描控制器-->
9   <context:component-scan base-package="cn.tulingxueyuan" use-default-filters="false">
10     <context:include-filter type="annotation" expression="org.springframework.stereotype.Controller"/>
11   </context:component-scan>
12   <!--静态资源的扫描-->
13   <mvc:default-servlet-handler></mvc:default-servlet-handler>
14   <!--动态资源的扫描-->
15   <mvc:annotation-driven></mvc:annotation-driven>
16   <!--配置视图管理器-->
17   <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
18     <property name="prefix" value="/WEB-INF/page/"></property>
19     <property name="suffix" value=".jsp"></property>
20   </bean>
21 </beans>

```

## spring-core.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xmlns:context="http://www.springframework.org/schema/context" xmlns:tx
   ="http://www.springframework.org/schema/tx"
5   xmlns:aop="http://www.springframework.org/schema/aop" xmlns:mybatis="http://mybatis.org/schema/mybatis-spring"
6   xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd http://www.springframework.org/schema/context https://www.springframework.org/schema/context/spring-context.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx.xsd http://www.springframework.org/schema/aop https://www.springframework.org/schema/aop/spring-aop.xsd http://mybatis.org/schema/mybatis-spring http://mybatis.org/schema/mybatis-spring.xsd">
7

```

```
8 <!--扫描所有除了controller包的其他包-->
9 <context:component-scan base-package="cn.tulingxueyuan">
10 <context:exclude-filter type="annotation" expression="org.springframework.stereotype.Controller"/>
11 </context:component-scan>
12
13
14 <!--引入外部属性资源文件-->
15 <context:property-placeholder location="classpath:db.properties"></context:property-placeholder>
16
17 <!--创建Druid数据源-->
18 <bean class="com.alibaba.druid.pool.DruidDataSource" id="dataSource">
19 <property name="username" value="${mysql.username}"></property>
20 <property name="password" value="${mysql.password}"></property>
21 <property name="url" value="${mysql.url}"></property>
22 <property name="driverClassName" value="${mysql.driverClassName}"></property>
23 </bean>
24
25 <!--声明式事务-->
26 <bean class="org.springframework.jdbc.datasource.DataSourceTransactionManager" id="transactionManager">
27 <property name="dataSource" ref="dataSource"></property>
28 </bean>
29
30 <!--基于驱动-->
31 <tx:annotation-driven transaction-manager="transactionManager"></tx:annotation-driven>
32
33
34 <!--用于声明事务切入的所有方法-->
35 <aop:config>
36 <aop:pointcut id="transactionCut" expression="execution(* cn.tulingxueyuan.service.impl.*(..))"/>
37 </aop:config>
38
39 <!--用来明确切点匹配到的方法哪些方法需要使用事务-->
40 <tx:advice id="advice" transaction-manager="transactionManager">
41 <tx:attributes>
42 <!--可以使用通配符-->
43 <tx:method name="add*" />
```



```

44 <tx:method name="update*" />
45 <tx:method name="delete*" />
46 <tx:method name="get*" read-only="true" propagation="SUPPORTS"></tx:met
hod>
47 <tx:method name="query*" read-only="true"></tx:method>
48 </tx:attributes>
49 </tx:advice>
50
51 <!--SqlSessionFactory-->
52 <bean class="org.mybatis.spring.SqlSessionFactoryBean" id="sqlSessionFa
ctory">
53 <!-- 指定spring中的数据源 -->
54 <property name="dataSource" ref="dataSource"></property>
55 <property name="configLocation" value="classpath:mybatis-config.xml">
</property>
56 <property name="mapperLocations" value="classpath:cn/tulingxueyuan/mapp
er/*.xml"></property>
57 </bean>
58
59 <!--将mapper接口交给spring管理-->
60 <mybatis:scan base-package="cn.tulingxueyuan.mapper"></mybatis:scan>
61
62 </beans>

```

## mybatis-config.xml

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE configuration
3 PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
4 "http://mybatis.org/dtd/mybatis-3-config.dtd">
5 <!--就是DOCTYPE后面对应的根节点-->
6 <configuration>
7
8
9 <!--mybatis的设置选项 可以改变mybatis运行时行为-->
10 <settings>
11 <setting name="mapUnderscoreToCamelCase" value="true"/>
12 </settings>
13
14 <!--类型别名可为 Java 类型设置一个缩写名字。 它仅用于 XML 配置，意在降低冗余
的全限定类名书写-->
15 <typeAliases>
16 <package name="cn.tulingxueyuan.pojo"/>

```

```
17 </typeAliases>
18
19 </configuration>
```

## db.properties

```
1 mysql.username = root
2 mysql.password=123456
3 mysql.url=jdbc:mysql://localhost:3306/mybatis
4 mysql.driverClassName=com.mysql.jdbc.Driver
```

## 生成 pojo\mapper接口\mapper映射文件

### TestController.java

```
1
2 /**
3  * @Author 徐庶 QQ:1092002729
4  * @Slogan 致敬大师，致敬未来的你
5  */
6 @RestController
7 public class EmpController {
8
9
10  @Autowired
11  IEmpService empService;
12
13  @RequestMapping("test")
14  public List<Emp> test(){
15  List<Emp> emps = empService.selectEmp();
16  return emps;
17  }
18 }
```