# SSM框架搭建

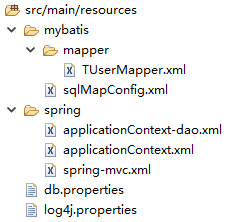
## pom.xml

|  |
| --- |
| <project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <groupId>org.leo</groupId>  <artifactId>ssm</artifactId>  <packaging>war</packaging>  <version>0.0.1-SNAPSHOT</version>  <name>ssm Maven Webapp</name>  <url>http://maven.apache.org</url>  <properties>  <spring.version>4.3.9.RELEASE</spring.version>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  </properties>  <dependencies>  <!-- 测试 -->  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  <version>4.12</version>  <scope>test</scope>  </dependency>  <!-- 日志适配器 -->  <dependency>  <groupId>org.apache.logging.log4j</groupId>  <artifactId>log4j-slf4j-impl</artifactId>  <version>2.8.2</version>  <scope>test</scope>  </dependency>  <!-- 日志实现 -->  <dependency>  <groupId>org.apache.logging.log4j</groupId>  <artifactId>log4j-core</artifactId>  <version>2.8.2</version>  </dependency>  <!-- aspectj框架包 -->  <dependency>  <groupId>org.aspectj</groupId>  <artifactId>aspectjweaver</artifactId>  <version>1.8.10</version>  </dependency>  <dependency>  <groupId>com.fasterxml.jackson.core</groupId>  <artifactId>jackson-core</artifactId>  <version>2.8.8</version>  </dependency>  <dependency>  <groupId>com.fasterxml.jackson.core</groupId>  <artifactId>jackson-databind</artifactId>  <version>2.8.8.1</version>  </dependency>  <!-- Spring依赖 -->  <!-- 1)spring核心依赖 -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-context</artifactId>  <version>${spring.version}</version>  </dependency>  <!--spring ioc依赖 -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-beans</artifactId>  <version>${spring.version}</version>  </dependency>  <!--2 spring web相关依赖 -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-webmvc</artifactId>  <version>${spring.version}</version>  <scope>runtime</scope>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-web</artifactId>  <version>${spring.version}</version>  </dependency>  <!—3 spring dao层依赖 -->  <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>  <!--4spring test相关依赖 -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-test</artifactId>  <version>${spring.version}</version>  </dependency>  <!-- mybatis -->  <dependency>  <groupId>org.mybatis</groupId>  <artifactId>mybatis</artifactId>  <version>3.4.3</version>  </dependency>  <dependency>  <groupId>org.mybatis</groupId>  <artifactId>mybatis-spring</artifactId>  <version>1.3.1</version>  </dependency>  <!-- 引入jstl包 -->  <dependency>  <groupId>jstl</groupId>  <artifactId>jstl</artifactId>  <version>1.2</version>  </dependency>  <!-- MySQL -->  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <version>5.1.42</version>  </dependency>  <!-- 数据库连接池 -->  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>druid</artifactId>  <version>1.0.31</version>  </dependency>  <dependency>  <groupId>org.apache.tomcat</groupId>  <artifactId>servlet-api</artifactId>  <version>6.0.37</version>  </dependency>  <dependency>  <groupId>org.apache.tomcat</groupId>  <artifactId>jsp-api</artifactId>  <version>6.0.37</version>  </dependency>  </dependencies>  <build>  <plugins>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-deploy-plugin</artifactId>  <version>2.7</version>  <configuration>  <uniqueVersion>false</uniqueVersion>  </configuration>  </plugin>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-eclipse-plugin</artifactId>  <version>2.8</version>  </plugin>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-compiler-plugin</artifactId>  <version>2.3.2</version>  <configuration>  <failOnError>true</failOnError>  <verbose>true</verbose>  <fork>true</fork>  <compilerArgument>-nowarn</compilerArgument>  <source>1.8</source>  <target>1.8</target>  <encoding>UTF-8</encoding>  </configuration>  </plugin>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-source-plugin</artifactId>  <version>2.1.2</version>  <executions>  <execution>  <id>attach-sources</id>  <goals>  <goal>jar</goal>  </goals>  </execution>  </executions>  </plugin>  </plugins>  </build>  <profiles>  <profile>  <id>jdk-1.8</id>  <!-- 另外一种激活方式 -->  <activation>  <activeByDefault>true</activeByDefault>  <jdk>1.8</jdk>  </activation>  <properties>  <maven.compiler.source>1.8</maven.compiler.source>  <maven.compiler.target>1.8</maven.compiler.target>  <maven.compiler.compilerVersion>1.8</maven.compiler.compilerVersion>  </properties>  </profile>  </profiles>  </project> |

## Web.xml

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <web-app version=*"2.5"* xmlns=*"http://java.sun.com/xml/ns/javaee"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance*  *http://www.springmodules.org/schema/cache/springmodules-cache.xsd*  *http://www.springmodules.org/schema/cache/springmodules-ehcache.xsd"*  xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee*  *http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd*  *"*>  <display-name>Archetype Created Web Application</display-name>  <servlet>  <servlet-name>dispatcherServlet</servlet-name>  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  <load-on-startup>1</load-on-startup>  <init-param>  <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>dispatcherServlet</servlet-name>  <url-pattern>/</url-pattern>  </servlet-mapping>  <!-- 加载spring容器 -->  <context-param>  <param-name>contextConfigLocation</param-name>  <param-value>classpath:spring/applicationContext.xml</param-value>  </context-param>  <listener>  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>  </listener>  <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>  </web-app> |

## 配置文件一览



## 配置文件

数据库配置文件db.properties：

|  |
| --- |
| ## JDBC set  jdbc.url=jdbc\:mysql\://192.168.56.102\:3306/ssm?useUnicode\=true&characterEncoding\=utf-8  jdbc.username=root  jdbc.password=xxxxxx |

日志配置文件log4j.properties：

|  |
| --- |
| #开发日志将在本地输出，并输出SQL  log4j.rootLogger=INFO,A1,DRF  log4j.appender.A1=org.apache.log4j.ConsoleAppender  log4j.appender.A1.layout=org.apache.log4j.PatternLayout  # log4j.appender.A1.layout.ConversionPattern=%d %5p [%t] (%F:%L) - %m%n  log4j.appender.A1.layout.ConversionPattern=%d %5p [%F:%L] : %m%n  log4j.appender.DRF=org.apache.log4j.DailyRollingFileAppender  log4j.appender.DRF.Threshold=INFO  log4j.appender.DRF.DatePattern='.'yyyy-MM-dd  log4j.appender.DRF.File=logs/ssm.log  log4j.appender.DRF.Append=true  log4j.appender.DRF.layout=org.apache.log4j.PatternLayout  log4j.appender.DRF.layout.ConversionPattern=[%-5p][%d{yyyyMMdd HH:mm:ss,SSS}][%C**{1}**:%L] %m%n  ###输出SQL  log4j.logger.com.ibatis=DEBUG  log4j.logger.com.ibatis.common.jdbc.SimpleDataSource=DEBUG  log4j.logger.com.ibatis.common.jdbc.ScriptRunner=DEBUG  log4j.logger.com.ibatis.sqlmap.engine.impl.SqlMapClientDelegate=DEBUG  log4j.logger.java.sql.Connection=DEBUG  log4j.logger.java.sql.Statement=DEBUG  log4j.logger.java.sql.PreparedStatement=DEBUG |

## MyBatis配置

sqlMapConfig.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>  <!-- 全局setting配置，根据需要添加 -->  <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"* />  <!-- 允许或不允许多种结果集从一个单独 的语句中返回(需要适合的驱动) -->  <setting name=*"multipleResultSetsEnabled"* value=*"true"* />  <!-- 使用列标签代替列名。 不同的驱动在这 方便表现不同。 参考驱动文档或充分测 试两种方法来决定所使用的驱动 -->  <setting name=*"useColumnLabel"* value=*"true"* />  <!-- 允许 JDBC 支持生成的键。 需要适合的 驱动。 如果设置为 true 则这个设置强制 生成的键被使用, 尽管一些驱动拒绝兼 容但仍然有效(比如Derby) -->  <setting name=*"useGeneratedKeys"* value=*"false"* />  <!-- 配置默认的执行器。SIMPLE 执行器没 有什么特别之处。REUSE 执行器重用 预处理语句。BATCH 执行器重用语句 和批量更新 -->  <setting name=*"defaultExecutorType"* value=*"SIMPLE"* />  <!-- 设置超时时间, 它决定驱动等待一个数 据库响应的时间 -->  <setting name=*"defaultStatementTimeout"* value=*"100"* />  <setting name=*"safeRowBoundsEnabled"* value=*"false"* />  <setting name=*"mapUnderscoreToCamelCase"* value=*"false"* />  <setting name=*"localCacheScope"* value=*"SESSION"* />  <setting name=*"jdbcTypeForNull"* value=*"OTHER"* />  <setting name=*"lazyLoadTriggerMethods"* value=*"equals,clone,hashCode,toString"* />  </settings>  </configuration> |

## Spring容器配置

applicationContext.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:p=*"http://www.springframework.org/schema/p"*  xmlns:tx=*"http://www.springframework.org/schema/tx"* xmlns:context=*"http://www.springframework.org/schema/context"*  xmlns:aop=*"http://www.springframework.org/schema/aop"*  xsi:schemaLocation=*"*  *http://www.springframework.org/schema/beans*  *http://www.springframework.org/schema/beans/spring-beans.xsd*  *http://www.springframework.org/schema/aop*  *http://www.springframework.org/schema/aop/spring-aop.xsd*  *http://www.springframework.org/schema/tx*  *http://www.springframework.org/schema/tx/spring-tx.xsd*  *http://www.springframework.org/schema/context*  *http://www.springframework.org/schema/context/spring-context.xsd"*>  <!-- 采用注释的方式配置bean -->  <context:annotation-config />  <!-- 配置要扫描的包 -->  <context:component-scan base-package=*"org.leo.ssm"*>  <context:exclude-filter type=*"annotation"*  expression=*"org.springframework.stereotype.Controller"* />  </context:component-scan>  <!-- proxy-target-class默认"false",更改为"ture"使用CGLib动态代理  <aop:aspectj-autoproxy proxy-target-class="true" /> -->  <import resource=*"applicationContext-dao.xml"* />  </beans> |

applicationContext-dao.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:p=*"http://www.springframework.org/schema/p"*  xmlns:util=*"http://www.springframework.org/schema/util"* xmlns:context=*"http://www.springframework.org/schema/context"*  xmlns:aop=*"http://www.springframework.org/schema/aop"* xmlns:tx=*"http://www.springframework.org/schema/tx"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans*  *http://www.springframework.org/schema/beans/spring-beans.xsd*  *http://www.springframework.org/schema/util*  *http://www.springframework.org/schema/util/spring-util.xsd*  *http://www.springframework.org/schema/context*  *http://www.springframework.org/schema/context/spring-context.xsd*  *http://www.springframework.org/schema/aop*  *http://www.springframework.org/schema/aop/spring-aop.xsd*  *http://www.springframework.org/schema/tx*  *http://www.springframework.org/schema/tx/spring-tx.xsd"*>  <!-- 读入配置属性文件 -->  <context:property-placeholder location=*"classpath:db.properties"* />  <!-- 基于Druid数据库链接池的数据源配置 -->  <bean id=*"dataSource"* class=*"com.alibaba.druid.pool.DruidDataSource"*  init-method=*"init"* destroy-method=*"close"*>  <!-- 基本属性driverClassName、 url、user、password -->  <property name=*"driverClassName"* value=*"com.mysql.jdbc.Driver"* />  <property name=*"url"* value=*"${jdbc.url}"* />  <property name=*"username"* value=*"${jdbc.username}"* />  <property name=*"password"* value=*"${jdbc.password}"* />  <!-- 配置初始化大小、最小、最大 -->  <!-- 通常来说，只需要修改initialSize、minIdle、maxActive -->  <property name=*"initialSize"* value=*"2"* />  <property name=*"minIdle"* value=*"2"* />  <property name=*"maxActive"* value=*"30"* />  <property name=*"testWhileIdle"* value=*"false"* />  <!-- 配置获取连接等待超时的时间 -->  <property name=*"maxWait"* value=*"5000"* />  <!-- 配置一个连接在池中最小生存的时间，单位是毫秒 -->  <property name=*"minEvictableIdleTimeMillis"* value=*"30000"* />  <!-- 配置间隔多久才进行一次检测，检测需要关闭的空闲连接，单位是毫秒 -->  <property name=*"timeBetweenEvictionRunsMillis"* value=*"60000"* />  <!-- 解密密码必须要配置的项 -->  <property name=*"filters"* value=*"config"* />  <property name=*"connectionProperties"* value=*"config.decrypt=false"* />  </bean>  <!-- 将数据源映射到sqlSessionFactory中 -->  <bean id=*"sqlSessionFactory"* class=*"org.mybatis.spring.SqlSessionFactoryBean"*>  <property name=*"dataSource"* ref=*"dataSource"* />  <property name=*"configLocation"* value=*"classpath:mybatis/sqlMapConfig.xml"* />  <property name=*"mapperLocations"* value=*"classpath:mybatis/mapper/\*Mapper.xml"*></property>  <property name=*"typeAliasesPackage"* value=*"org.leo.ssm.entity.bo"* />  </bean>  <bean class=*"org.mybatis.spring.mapper.MapperScannerConfigurer"*>  <property name=*"basePackage"* value=*"org.leo.ssm.mapper"* />  </bean>  <!-- SqlSession模板类实例 -->  <bean id=*"sessionTemplate"* class=*"org.mybatis.spring.SqlSessionTemplate"*  destroy-method=*"close"*>  <constructor-arg index=*"0"* ref=*"sqlSessionFactory"* />  </bean>  <!--======= 事务配置 Begin ================= -->  <!-- 事务管理器（由Spring管理MyBatis的事务） -->  <bean id=*"transactionManager"*  class=*"org.springframework.jdbc.datasource.DataSourceTransactionManager"*>  <property name=*"dataSource"* ref=*"dataSource"* />  </bean>  <!-- 使用annotation定义事务 -->  <tx:annotation-driven transaction-manager=*"transactionManager"* />  <!--======= 事务配置 End =================== -->  </beans> |

## EasyUI与服务端通过Ajax交互

EasyUI页面，users.html（不含引入代码）：

|  |
| --- |
| <body>  <table id="dg"></table>  <script type="text/javascript">  $(function(){  $.ajax({  url: "http://127.0.0.1:8080/ssm/user/findUsers.do",  type: "post",  dataType: "json",  data:JSON.stringify({"id":1,"username":"1","gender":1}),  contentType:'application/json;charset=UTF-8',  success: function (data) {  $("#dg").datagrid({  title: '用户列表',  width: 700,  height: 250,  fitColumns: true,  singleSelect: true,  loadMsg:"努力加载中",  emptyMsg:"没有记录",  columns:[[  {field:'id',title:'主键',width:80},  {field:'username',title:'用户名',width:120},  {field:'gender',title:'性别',width:80,  formatter:function(value,row){  //return row.gender=='0'?'女':'男';  return value==0?'女':'男';  }},  {field:'intime',title:'创建时间',width:80,  formatter : function(value){  return getTime(String(value).substring(0,10));  }  }]]  });  $("#dg").datagrid("loadData", data.rows); //动态取数据  }  });  });  //时间戳格式化日期时间  function getTime(ts) {  var t, y, m, d, h, i, s;  t = ts ? new Date(ts \* 1000) : new Date();  y = t.getFullYear();  m = t.getMonth() + 1;  d = t.getDate();  h = t.getHours();  i = t.getMinutes();  s = t.getSeconds();  // 可根据需要在这里定义时间格式  return y + '-' + (m < 10 ? '0' + m : m) + '-' + (d < 10 ? '0' + d : d) + ' ' + (h < 10 ? '0' + h : h) + ':' + (i < 10 ? '0' + i : i) + ':' + (s < 10 ? '0' + s : s);  }  </script>  </body> |