# Maven构建分布式工程

## 总目录

common-parent：Maven父工程，主要在pom存放全量依赖jar。

common-config：公共配置工程，放置公用的配置文件，比如数据库访问、Dubbo路径等。

common-core：公共core，放置Dao层需要的一些公共类，本示例暂无。

common-base：公共实体类，放置静态参数类、工具类等。

common-entity：BO&VO，有时候一个业务会使用其他业务的BO、VO，所以讲所有模块的实体类都集中放置。注意：所有类必须序列化。

common-web：公共WEB，放置Web相关的一些类，比如分页类、Session处理相关类。本示例暂无。

controller-user：对外（HTML、APP）提供HTTP接口。

interface-user：Service的接口，供controller-user引入，供service-user实现。

service-user：Service的实现，实现interface-user的接口，并通过MyBatis的Mapper与数据库交互。

## 配置文件

本节会列出所有工程的pom、xml、properties文件。

### common-parent

#### 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/xsd/maven-4.0.0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <groupId>org.leo.common</groupId>  <artifactId>common-parent</artifactId>  <version>1.0-SNAPSHOT</version>  <packaging>pom</packaging>  <name>common-parent</name>  <url>http://maven.apache.org</url>  <description>所有工程的父工程，主要设置依赖jar</description>  <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  <!-- common version -->  <common-config.version>1.0-SNAPSHOT</common-config.version>  <common-core.version>1.0-SNAPSHOT</common-core.version>  <common-base.version>1.0-SNAPSHOT</common-base.version>  <common-entity.version>1.0-SNAPSHOT</common-entity.version>  <common-web.version>1.0-SNAPSHOT</common-web.version>  <!-- user version -->  <controller-user.version>1.0-SNAPSHOT</controller-user.version>  <interface-user.version>1.0-SNAPSHOT</interface-user.version>  <service-user.version>1.0-SNAPSHOT</service-user.version>  <!-- spring version -->  <spring.version>4.3.9.RELEASE</spring.version>  </properties>  <dependencyManagement>  <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-aop</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-context-support</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>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-instrument</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-instrument-tomcat</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-jms</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-orm</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-oxm</artifactId>  <version>${spring.version}</version>  </dependency>  <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坐标依赖 -->  <!--3 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>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-webmvc-portlet</artifactId>  <version>${spring.version}</version>  </dependency>  <!--4spring dao层依赖 -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-jdbc</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-tx</artifactId>  <version>${spring.version}</version>  </dependency>  <!--5spring test相关依赖 -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-test</artifactId>  <version>${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>  <!-- zookeeper -->  <dependency>  <groupId>org.apache.zookeeper</groupId>  <artifactId>zookeeper</artifactId>  <version>3.4.10</version>  </dependency>  <dependency>  <groupId>com.101tec</groupId>  <artifactId>zkclient</artifactId>  <version>0.10</version>  </dependency>  <dependency>  <groupId>org.jboss.netty</groupId>  <artifactId>netty</artifactId>  <version>3.2.5.Final</version>  </dependency>  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>dubbo</artifactId>  <version>2.5.3</version>  </dependency>  </dependencies>  </dependencyManagement>  <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>  </project> |

### common-config

#### 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/xsd/maven-4.0.0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>org.leo.common</groupId>  <artifactId>common-parent</artifactId>  <version>1.0-SNAPSHOT</version>  <relativePath>../common-parent</relativePath>  </parent>  <description>存放配置文件</description>  <artifactId>common-config</artifactId>  <version>${common-config.version}</version>  <packaging>jar</packaging>  <name>common-config</name>  <url>http://maven.apache.org</url>  <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  </properties>  <build>  <finalName>common-config</finalName>  <resources>  <!-- 指定 src/main/resources下所有文件及文件夹为资源文件 -->  <resource>  <directory>src/main/resources</directory>  <targetPath>${project.build.directory}/classes</targetPath>  <includes>  <include>\*\*/\*</include>  </includes>  <filtering>true</filtering>  </resource>  </resources>  </build>  </project> |

配置文件都放在src/main/resources目录下

#### db.properties

|  |
| --- |
| ## JDBC set  jdbc.url=jdbc\:mysql\://192.168.56.102\:3306/ssm?useUnicode\=true&characterEncoding\=utf-8&useSSL=false  jdbc.username=root  jdbc.password=MyNewPwd4! |

#### service.properties

|  |
| --- |
| ## Dubbo  dubbo.registry.address=192.168.56.102:2181 |

### common-base

#### 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/xsd/maven-4.0.0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>org.leo.common</groupId>  <artifactId>common-parent</artifactId>  <version>1.0-SNAPSHOT</version>  <relativePath>../common-parent</relativePath>  </parent>  <description>存放公共类、静态类、工具类等</description>  <artifactId>common-base</artifactId>  <version>${common-base.version}</version>  <packaging>jar</packaging>  <name>common-base</name>  <url>http://maven.apache.org</url>  <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  </properties>  </project> |

### common-entity

#### 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/xsd/maven-4.0.0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>org.leo.common</groupId>  <artifactId>common-parent</artifactId>  <version>1.0-SNAPSHOT</version>  <relativePath>../common-parent</relativePath>  </parent>  <description>存放BO/VO</description>  <artifactId>common-entity</artifactId>  <version>${common-entity.version}</version>  <packaging>jar</packaging>  <name>common-entity</name>  <url>http://maven.apache.org</url>  <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  </properties>  </project> |

### common-core

#### 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/xsd/maven-4.0.0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>org.leo.common</groupId>  <artifactId>common-parent</artifactId>  <version>1.0-SNAPSHOT</version>  <relativePath>../common-parent</relativePath>  </parent>  <description>存放Dao层公共类</description>  <artifactId>common-core</artifactId>  <version>${common-core.version}</version>  <packaging>jar</packaging>  <name>common-core</name>  <url>http://maven.apache.org</url>  <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  </properties>  </project> |

### common-web

#### 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/xsd/maven-4.0.0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>org.leo.common</groupId>  <artifactId>common-parent</artifactId>  <version>1.0-SNAPSHOT</version>  <relativePath>../common-parent</relativePath>  </parent>  <description>存放WEB Controller层公共类</description>  <artifactId>common-web</artifactId>  <version>${common-web.version}</version>  <packaging>jar</packaging>  <name>common-web</name>  <url>http://maven.apache.org</url>  <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  </properties>  <dependencies>  <dependency>  <groupId>org.leo</groupId>  <artifactId>common-base</artifactId>  <version>${common-base.version}</version>  </dependency>  <dependency>  <groupId>org.apache.tomcat</groupId>  <artifactId>servlet-api</artifactId>  <scope>provided</scope>  </dependency>  <dependency>  <groupId>org.apache.tomcat</groupId>  <artifactId>jsp-api</artifactId>  <scope>provided</scope>  </dependency>  </dependencies>  </project> |

### Interface-user

#### 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/xsd/maven-4.0.0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>org.leo.common</groupId>  <artifactId>common-parent</artifactId>  <version>1.0-SNAPSHOT</version>  <relativePath>../common-parent</relativePath>  </parent>  <description>用户服务的接口</description>  <groupId>org.leo.interface</groupId>  <artifactId>Interface-user</artifactId>  <version>${interface-user.version}</version>  <packaging>jar</packaging>  <name>Interface-user</name>  <url>http://maven.apache.org</url>  <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  </properties>  <dependencies>  <dependency>  <groupId>org.leo.common</groupId>  <artifactId>common-entity</artifactId>  <version>${common-entity.version}</version>  </dependency>  </dependencies>  </project> |

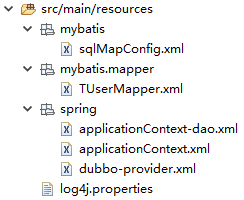
### service-user

#### 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>  <parent>  <groupId>org.leo.common</groupId>  <artifactId>common-parent</artifactId>  <version>1.0-SNAPSHOT</version>  <relativePath>../common-parent</relativePath>  </parent>  <description>用户服务的Service实现</description>  <groupId>org.leo.service</groupId>  <artifactId>service-user</artifactId>  <version>${service-user.version}</version>  <packaging>jar</packaging>  <name>service-user</name>  <url>http://maven.apache.org</url>  <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  <log4j.leve>debug</log4j.leve>  <log4j.ale>debug</log4j.ale>  </properties>  <dependencies>  <dependency>  <groupId>org.leo.common</groupId>  <artifactId>common-base</artifactId>  <version>${common-base.version}</version>  </dependency>  <dependency>  <groupId>org.leo.common</groupId>  <artifactId>common-config</artifactId>  <version>${common-config.version}</version>  </dependency>  <dependency>  <groupId>org.leo.common</groupId>  <artifactId>common-core</artifactId>  <version>${common-core.version}</version>  </dependency>  <dependency>  <groupId>org.leo.common</groupId>  <artifactId>common-entity</artifactId>  <version>${common-entity.version}</version>  </dependency>  <dependency>  <groupId>org.leo.interface</groupId>  <artifactId>Interface-user</artifactId>  <version>${interface-user.version}</version>  </dependency>  <!-- 测试 -->  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  <scope>test</scope>  </dependency>  <!-- 日志适配器 -->  <dependency>  <groupId>org.apache.logging.log4j</groupId>  <artifactId>log4j-slf4j-impl</artifactId>  <scope>test</scope>  </dependency>  <!-- 日志实现 -->  <dependency>  <groupId>org.apache.logging.log4j</groupId>  <artifactId>log4j-core</artifactId>  </dependency>  <!-- aspectj框架包 -->  <dependency>  <groupId>org.aspectj</groupId>  <artifactId>aspectjweaver</artifactId>  </dependency>  <dependency>  <groupId>com.fasterxml.jackson.core</groupId>  <artifactId>jackson-core</artifactId>  </dependency>  <dependency>  <groupId>com.fasterxml.jackson.core</groupId>  <artifactId>jackson-databind</artifactId>  </dependency>  <!-- Spring依赖 -->  <!-- 1)spring核心依赖 -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-context</artifactId>  </dependency>  <!--spring ioc依赖 -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-beans</artifactId>  </dependency>  <!--2 spring坐标依赖 -->  <!--3 spring web相关依赖 -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-webmvc</artifactId>  <scope>runtime</scope>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-web</artifactId>  </dependency>  <!--4spring dao层依赖 -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-jdbc</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-tx</artifactId>  </dependency>  <!--5spring test相关依赖 -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-test</artifactId>  </dependency>  <!-- mybatis -->  <dependency>  <groupId>org.mybatis</groupId>  <artifactId>mybatis</artifactId>  </dependency>  <dependency>  <groupId>org.mybatis</groupId>  <artifactId>mybatis-spring</artifactId>  </dependency>  <!-- MySQL -->  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  </dependency>  <!-- 数据库连接池 -->  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>druid</artifactId>  </dependency>  <!-- dubbo 需要的jar start -->  <dependency>  <groupId>org.jboss.netty</groupId>  <artifactId>netty</artifactId>  </dependency>  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>dubbo</artifactId>  <exclusions>  <exclusion>  <groupId>org.springframework</groupId>  <artifactId>spring</artifactId>  </exclusion>  </exclusions>  </dependency>  <dependency>  <groupId>org.apache.zookeeper</groupId>  <artifactId>zookeeper</artifactId>  </dependency>  <dependency>  <groupId>com.101tec</groupId>  <artifactId>zkclient</artifactId>  </dependency>  <!-- dubbo 需要的jar end -->  </dependencies>  <build>  <finalName>service-user</finalName>  <resources>  <resource>  <targetPath>${project.build.directory}/classes</targetPath>  <directory>src/main/resources</directory>  <filtering>true</filtering>  <includes>  <include>\*\*/\*.xml</include>  <include>\*\*/\*.properties</include>  </includes>  </resource>  </resources>  </build>  </project> |

#### web.xml

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xmlns=*"http://java.sun.com/xml/ns/javaee"* xmlns:web=*"http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*  xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*  id=*"gw-service-merchant"* version=*"2.5"*>  <display-name>用户服务</display-name>  <!-- 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>  </web-app> |



#### dubbo-provider.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:dubbo=*"http://code.alibabatech.com/schema/dubbo"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans*  *http://www.springframework.org/schema/beans/spring-beans.xsd*  *http://code.alibabatech.com/schema/dubbo*  *http://code.alibabatech.com/schema/dubbo/dubbo.xsd"*>  <!-- 提供方应用信息，用于计算依赖关系 -->  <dubbo:application name=*"leo-service-user"* />  <!-- 使用zookeeper注册中心暴露服务地址 -->  <dubbo:registry protocol=*"zookeeper"* address=*"${dubbo.registry.address}"* />  <!-- 用dubbo协议在20880端口暴露服务 -->  <dubbo:protocol name=*"dubbo"* port=*"20880"* />  <!-- 用户服务接口 -->  <dubbo:service interface=*"org.leo.ssm.service.UserService"*  ref=*"userService"* />  </beans> |

#### 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> |

#### 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"*>  <!-- 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>  <!-- mapper扫描器 <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">  扫描包路径，如果需要扫描多个包，中间使用半角逗号隔开 <property name="basePackage" value="org.leo.ssm.mapper"></property>  <property name="sqlSessionFactoryBeanName" value="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> |

#### 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 />  <!-- 读入配置属性文件 -->  <context:property-placeholder location=*"classpath:db.properties,classpath:service.properties"* />  <!-- 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"* />  <import resource=*"dubbo-provider.xml"* />  </beans> |

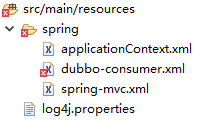
### controller-user

#### 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/xsd/maven-4.0.0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>org.leo.common</groupId>  <artifactId>common-parent</artifactId>  <version>1.0-SNAPSHOT</version>  <relativePath>../common-parent</relativePath>  </parent>  <description>用户服务的Controller</description>  <groupId>org.leo.controller</groupId>  <artifactId>controller-user</artifactId>  <version>${controller-user.version}</version>  <packaging>war</packaging>  <name>controller-user</name>  <url>http://maven.apache.org</url>  <build>  <finalName>controller-user</finalName>  <resources>  <resource>  <targetPath>${project.build.directory}/classes</targetPath>  <directory>src/main/resources</directory>  <filtering>true</filtering>  <includes>  <include>\*\*/\*.xml</include>  <include>\*\*/\*.properties</include>  </includes>  </resource>  </resources>  </build>  <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  </properties>  <dependencies>  <dependency>  <groupId>org.leo.common</groupId>  <artifactId>common-config</artifactId>  <version>${common-config.version}</version>  </dependency>  <dependency>  <groupId>org.leo.common</groupId>  <artifactId>common-entity</artifactId>  <version>${common-entity.version}</version>  </dependency>  <dependency>  <groupId>org.leo.common</groupId>  <artifactId>common-web</artifactId>  <version>${common-web.version}</version>  </dependency>  <dependency>  <groupId>org.leo.interface</groupId>  <artifactId>Interface-user</artifactId>  <version>${interface-user.version}</version>  </dependency>  <!-- Spring Dependency Begin -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-aop</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-aspects</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-beans</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-context</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-context-support</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-core</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-expression</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-instrument</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-instrument-tomcat</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-jdbc</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-jms</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-orm</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-oxm</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-test</artifactId>  <scope>test</scope>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-tx</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-web</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-webmvc</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-webmvc-portlet</artifactId>  </dependency>  <!-- Spring Dependency End -->  <dependency>  <groupId>com.fasterxml.jackson.core</groupId>  <artifactId>jackson-core</artifactId>  </dependency>  <dependency>  <groupId>com.fasterxml.jackson.core</groupId>  <artifactId>jackson-databind</artifactId>  </dependency>  <!-- dubbo 需要的jar start -->  <dependency>  <groupId>org.jboss.netty</groupId>  <artifactId>netty</artifactId>  </dependency>  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>dubbo</artifactId>  <exclusions>  <exclusion>  <groupId>org.springframework</groupId>  <artifactId>spring</artifactId>  </exclusion>  </exclusions>  </dependency>  <dependency>  <groupId>org.apache.zookeeper</groupId>  <artifactId>zookeeper</artifactId>  </dependency>  <dependency>  <groupId>com.101tec</groupId>  <artifactId>zkclient</artifactId>  </dependency>  <!-- dubbo 需要的jar end -->  </dependencies>  </project> |

#### web.xml

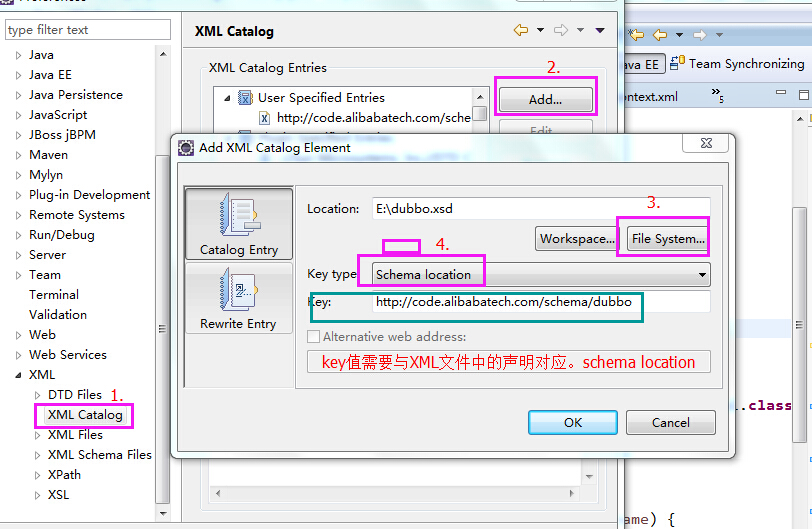
|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <web-app version=*"2.5"* xmlns=*"http://java.sun.com/xml/ns/javaee"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance*  *http://www.springmodules.org/schema/cache/springmodules-cache.xsd*  *http://www.springmodules.org/schema/cache/springmodules-ehcache.xsd"*  xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee*  *http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd*  *"*>  <display-name>用户HTTP接口</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> |



#### dubbo-consumer.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:dubbo=*"http://code.alibabatech.com/schema/dubbo"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans*  *http://www.springframework.org/schema/beans/spring-beans.xsd*  *http://code.alibabatech.com/schema/dubbo*  *http://code.alibabatech.com/schema/dubbo/dubbo.xsd"*>  <!-- 消费方应用名，用于计算依赖关系，不是匹配条件，不要与提供方一样 -->  <dubbo:application name=*"leo-controller-user"* />  <!-- 使用zookeeper注册中心暴露服务地址 -->  <!-- 注册中心地址 -->  <dubbo:registry protocol=*"zookeeper"* address=*"${dubbo.registry.address}"* />  <!-- 用户服务接口 -->  <dubbo:reference interface=*"org.leo.ssm.service.UserService"*  id=*"userService"* check=*"false"* />  </beans> |

注意：这里经常会报错，但是不影响使用，如想解决红叉问题，可以从引入的dubbo.jar包中解压缩出dubbo.xsd，在Eclipse中进行如下配置：



#### spring-mvc.xml

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <beans xmlns=*"http://www.springframework.org/schema/beans"*  xmlns:mvc=*"http://www.springframework.org/schema/mvc"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  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/tx*  *http://www.springframework.org/schema/tx/spring-tx.xsd*  *http://www.springframework.org/schema/mvc*  *http://www.springframework.org/schema/mvc/spring-mvc.xsd*  *http://www.springframework.org/schema/context*  *http://www.springframework.org/schema/context/spring-context.xsd"*>  <mvc:annotation-driven>  <mvc:message-converters>  <bean  class=*"org.springframework.http.converter.json.MappingJackson2HttpMessageConverter"*>  <property name=*"supportedMediaTypes"*>  <list>  <value>application/json;charset=UTF-8</value>  </list>  </property>  </bean>  </mvc:message-converters>  </mvc:annotation-driven>  <context:component-scan base-package=*"org.leo.ssm.controller"*>  <context:include-filter type=*"annotation"*  expression=*"org.springframework.stereotype.Controller"* />  <context:exclude-filter type=*"annotation"*  expression=*"org.springframework.stereotype.Service"* />  </context:component-scan>  <bean id=*"jspViewResolver"*  class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>  <property name=*"viewClass"*  value=*"org.springframework.web.servlet.view.JstlView"* />  <property name=*"prefix"* value=*"/WEB-INF/jsp/"* />  <property name=*"suffix"* value=*".jsp"* />  </bean>  </beans> |

#### 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 />  <!-- 读入配置属性文件 -->  <context:property-placeholder location=*"classpath:service.properties"* />  <!-- 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=*"dubbo-consumer.xml"* />  </beans> |

## 开发环境中运行

### 1、启动服务器相关环境

先启动服务器的ZK服务、Dubbo管控台Tomcat、MySQL数据库。

### 2、启动应用提供者

service-user中创建类DubboProvider，代码如下：

|  |
| --- |
| package org.leo.junit.common;  import org.apache.commons.logging.Log;  import org.apache.commons.logging.LogFactory;  import org.springframework.context.support.ClassPathXmlApplicationContext;  public class DubboProvider {  private static final Log log = LogFactory.getLog(DubboProvider.class);  public static void main(String[] args) {  try {  ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("classpath:spring/applicationContext.xml");  context.start();  } catch (Exception e) {  log.error("== DubboProvider context start error:",e);  }  synchronized (DubboProvider.class) {  while (true) {  try {  DubboProvider.class.wait();  } catch (InterruptedException e) {  log.error("== synchronized error:",e);  }  }  }  }  } |

直接运行，使服务注册到ZK。

### 3、部署接口war

Maven Install将controller-user打成war，部署到Tomcat并启动。

通过静态页面发送Ajax请求，结果如下：

