**1.项目结构：**

服务提供者ServiceImpl模块和服务消费者Controller模块，打成war包，

**Service打成war包的原因**：

打成war包，相当于一个独立的工程，加载一次服务，即可为多个消费者提供服务。否则的话，每次消费者调用服务，都需要加载。效率低。

**2.pom文件依赖：**

controller模块（war包）依赖serviceImpl模块，

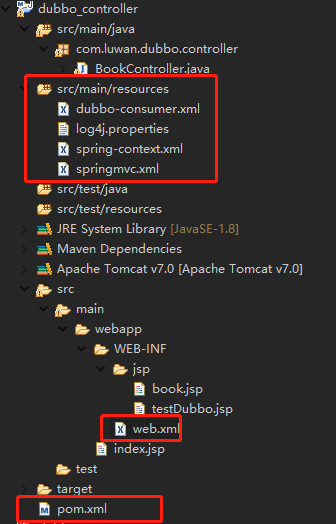
serviceImpl（war包）依赖Service，

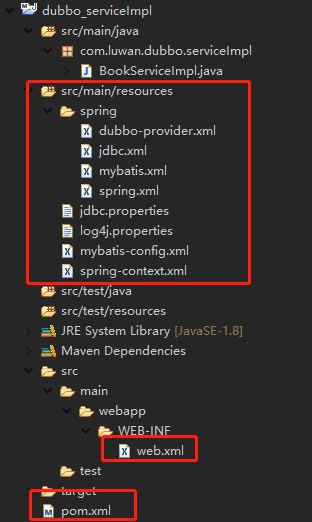
service（jar包）依赖common（bean和dao）

common（jar包）的pom配置Spring、SpringMVC、Mybatis、Redis、Mysql等等，版本控制由parent管理。

parent（pom包）统一管理依赖的版本号。

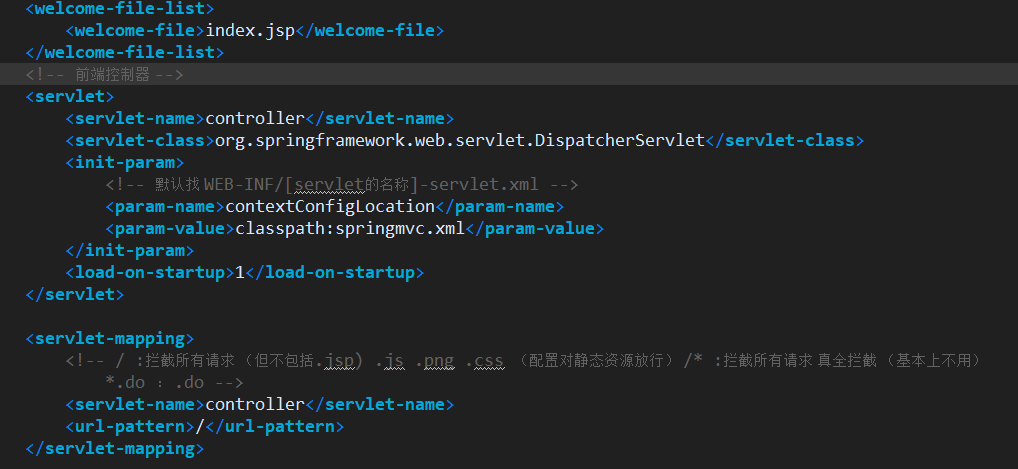
**3.配置文件：**





**controller配置文件：**

**web.xml中配置欢迎页、前端控制器Servlet、注入springmvc配置等**



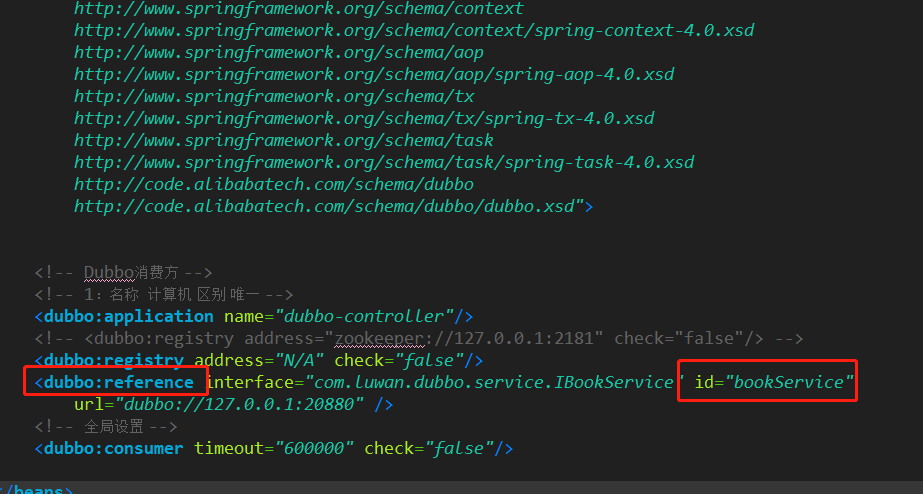
**SpringMVC的配置，扫描@Controller注解，映射器，适配器，导入服务消费者配置文件。**

**log4j.properties配置**

<mvc:annotation-driven>会自动注册RequestMappingHandlerMapping与RequestMappingHandlerAdapter两个Bean,这是Spring MVC为@Controller分发请求所必需的，并且提供了数据绑定支持，@NumberFormatannotation支持，@DateTimeFormat支持,@Valid支持读写XML的支持（JAXB）和读写JSON的支持（默认Jackson）等功能。

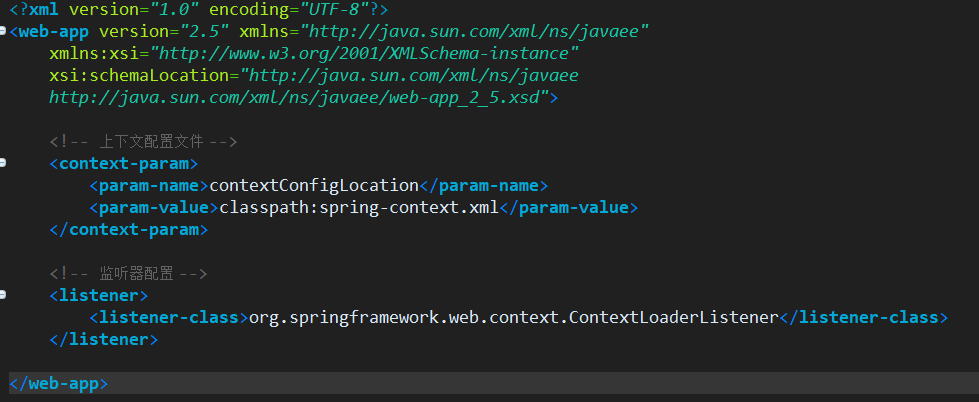


**服务消费者dubbo-consumer.xml**，消费的服务关联到服务提供者。



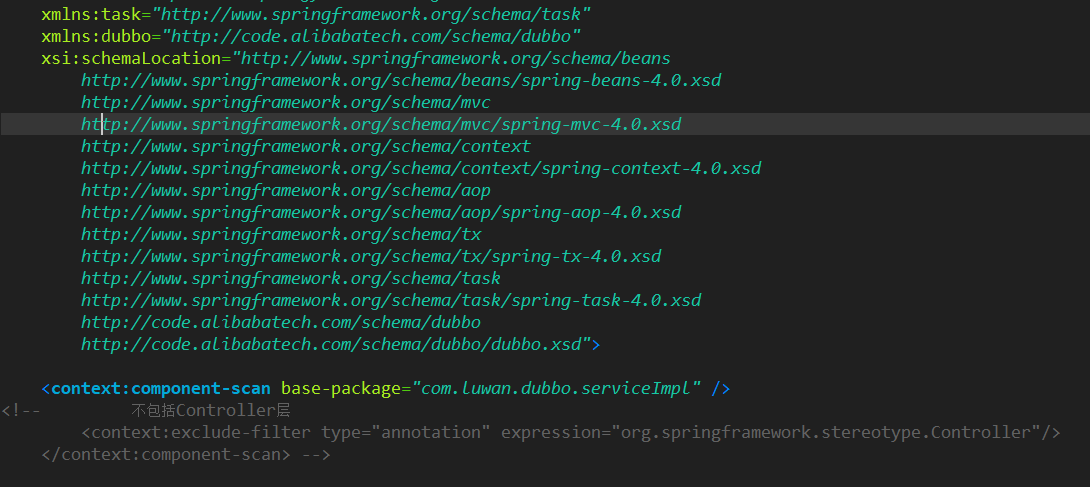
**Service配置文件：**

**web.xml**中配置Spring框架的context上下文配置加载，监听器配置等。

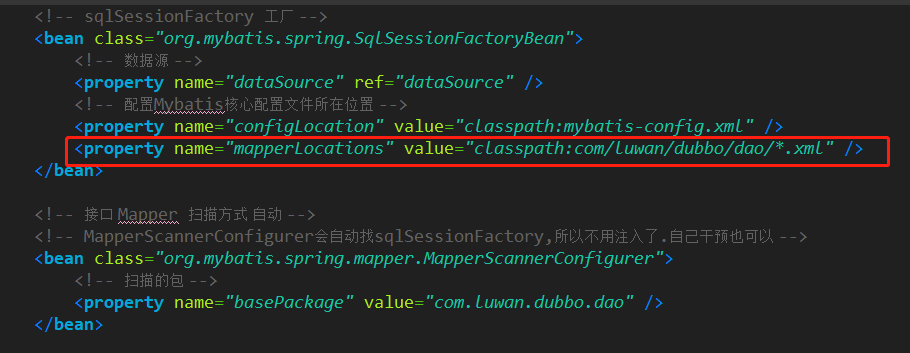


Spring配置：spring-context.xml，导入spring文件夹中的配置。将Spring整合mybaits、dubbo等的配置都放入一个文件夹中（如spring文件夹）。在spring文件夹中配置spring、mybatis、jdbc、dubbo-provicder等配置。

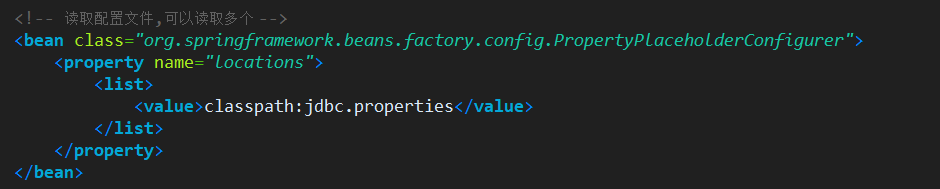
spring配置服务注解扫描。



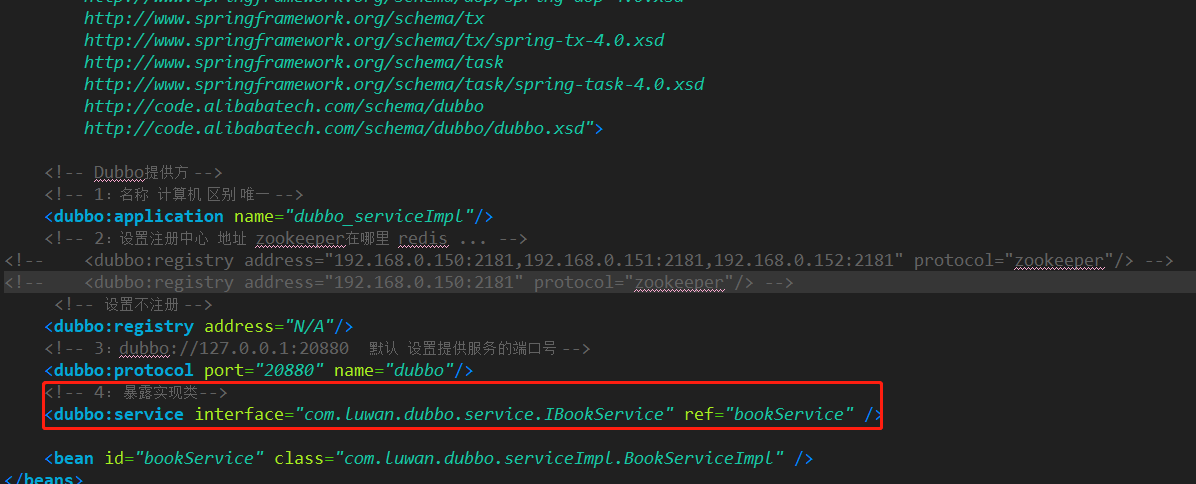
mybatis配置中，**注意：若mapper和dao接口不在同一文件夹下（改过名字和路径），则需要配置mapper.xml的包路径。**



配置jdbc.xml，在其中导入jdbc.properties，配置aop事务管理等。



服务消费者配置dubbo-provider.xml，**注意：暴露服务的配置，必须是暴露接口，不能是实现类。关联（ref）到实现类的注解（@Service）名称。否则项目启动时会出现serviceImpl无法autowired注入的报错。**



**4.启动和访问**

服务提供者和消费者分别**部署到不同的服务器（tomcat）上（在一台电脑上部署时记得修改端口号等配置，保证端口不冲突）**，项目启动时，**先启动服务提供者Service，后启动消费者。**

访问的时候，请求地址里别忘了带上**项目名称和参数**。

