分别为dubbo 2.5.7、2.6.2、2.6.8、2.7.3 的靶场

注意：如果访问页面出现**timeout** 字样，启动时加

**-Ddubbo.consumer.timeout=30000**

## 部署方式

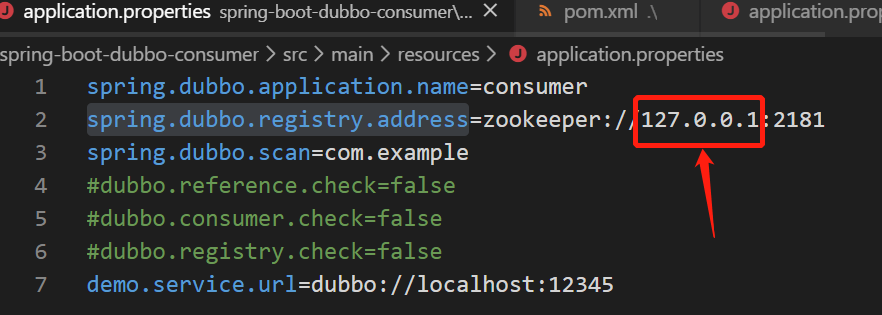
### 2.5.7

1. 使用 dubbo2.5.7\apache-zookeeper-3.5.6\bin 目录下的zkServer.cmd 或zkServer.sh 启动zookeeper服务

2. 部署前确定启动zookeeper服务器的ip，修改三个war包中的

**WEB-INF\classes\application.properties** 配置文件中的

**spring.dubbo.registry.address** 配置项，修改127.0.0.1为zookeeper服务器ip



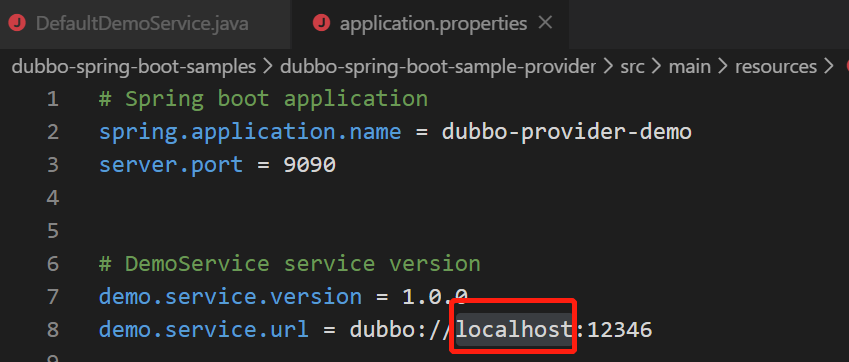
3. 将 spring-boot-dubbo-consumer.war 、spring-boot-dubbo-provider.war、 spring-boot-dubbo-provider2.war 部署在tomcat上（最好分开部署在不同服务器上）

4. **依次启动** spring-boot-dubbo-provider2.war 、spring-boot-dubbo-provider.war、spring-boot-dubbo-consumer.war 所在的tomcat

### 2.6.2

1.使用java -jar dubbo-spring-boot-provider2-2.6.2.jar启动provider2

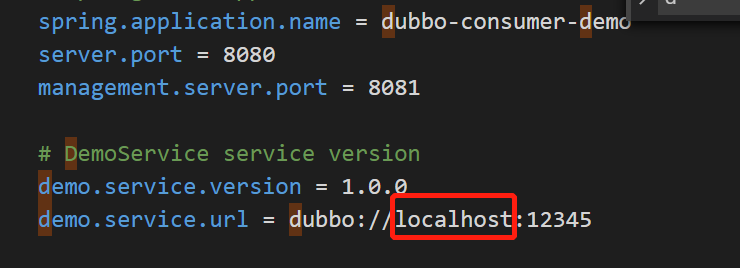
2.修改 dubbo-spring-boot-provider-2.6.2.jar 中 BOOT-INF\classes\application.properties 配置文件中的 demo.service.url 配置项，将localhost 改为启动provider2的服务器ip



然后使用java -jar 命令启动provider

3.修改dubbo-spring-boot-consumer-2.6.2.jar 中的 application.properties配置文件中的

demo.service.url 配置项，改为启动provider的ip



然后使用java -jar 命令启动

### 2.6.8

1.使用java -jar dubbo-spring-boot-provider2-2.6.8.jar启动provider2

2.修改 dubbo-spring-boot-provider-2.6.8.jar 中 BOOT-INF\classes\application.properties 配置文件中的 demo.service.url 配置项，将localhost 改为启动provider2的服务器ip

然后使用java -jar 命令启动provider

3.修改dubbo-spring-boot-consumer-2.6.8.jar 中的 application.properties配置文件中的

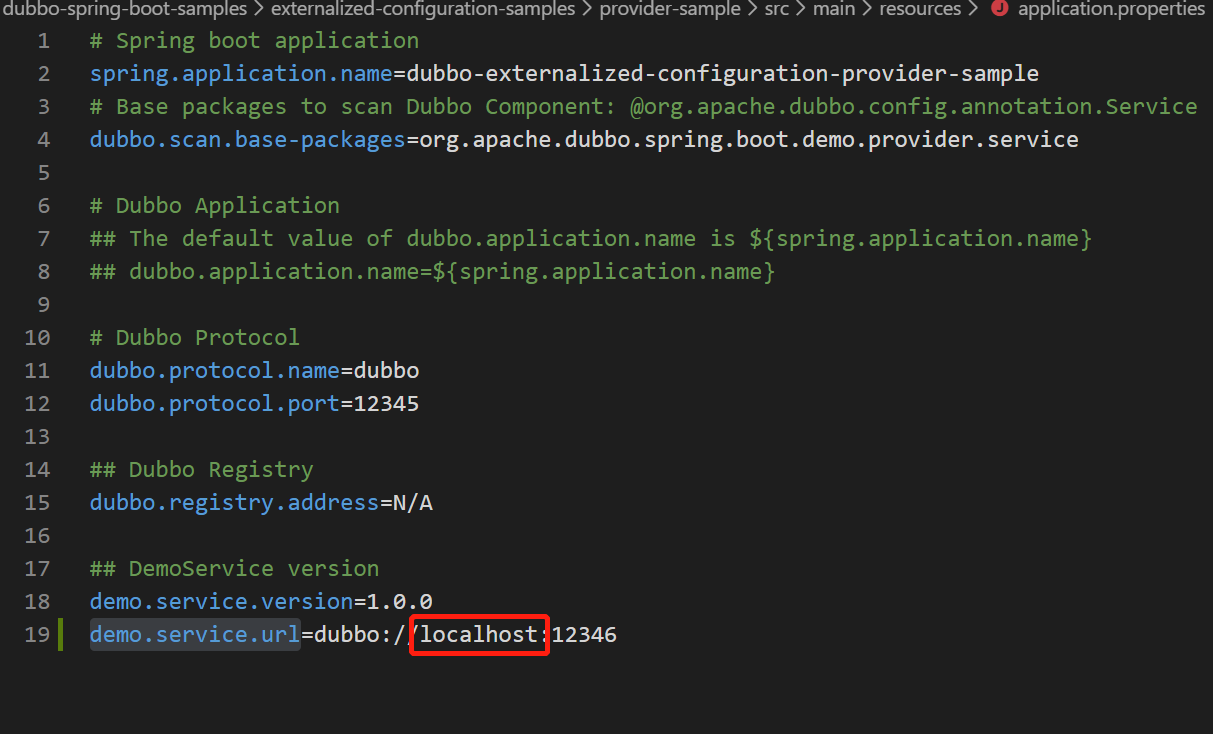
demo.service.url 配置项，改为启动provider的ip

然后使用java -jar 命令启动

### 2.7.3

1.使用java -jar dubbo-spring-boot-provider2-2.7.3.jar启动provider2

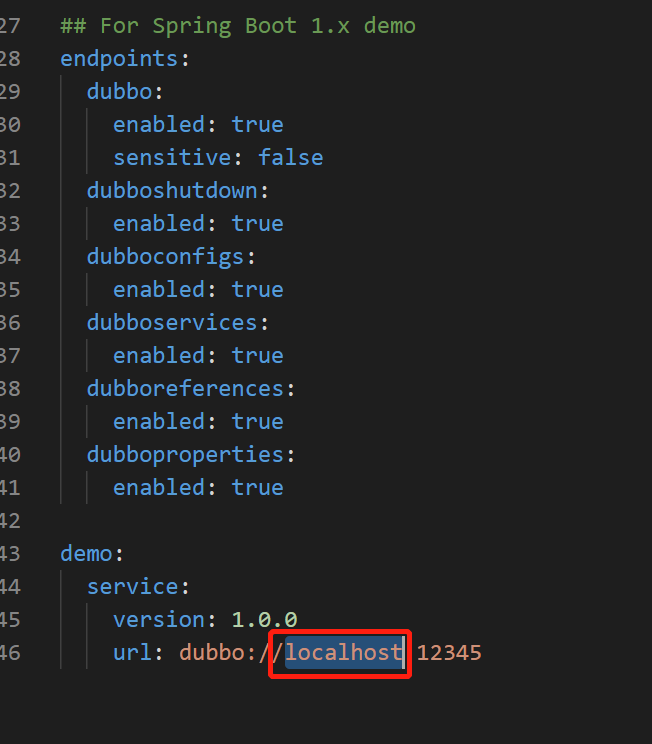
2.修改 dubbo-spring-boot-provider-2.7.3.jar 中 BOOT-INF\classes\application.properties 配置文件中的 demo.service.url 配置项，将localhost 改为启动provider2的服务器ip



然后使用java -jar 命令启动provider

3.修改dubbo-spring-boot-consumer-2.7.3.jar 中的 application.yml 配置文件中的

demo.service.url 配置项，改为启动provider的ip



然后使用java -jar 命令启动

## 访问地址：

1. <http://localhost:8080/stringVuln?name=calc>
2. <http://localhost:8080/pojoVuln?name=calc>
3. <http://localhost:8080/multiPojoVuln?name=calc>
4. <http://localhost:8080/mixPojoVuln?name=calc&name2=calc>

访问时需替换localhost为consumer所在的 ip及端口号

漏洞说明：

StringVuln 参数为简单String 类型

PojoVuln 参数为 简单Pojo对象

MultiPojoVuln 参数为 含有100个成员的Pojo对象

MixPojoVuln 漏洞为两个参数的拼接

测试同学只要看有没有命令执行或反射型跨站脚本漏洞出现即可，数据流及漏洞准确性 安全同学会进行验证