**Docker 容器启动Dubbo 服务**

|  |  |  |
| --- | --- | --- |
| 作者 | 日期 |  |
| 姜鹏 | 2018/09/06 |  |

微信号：comeon\_betty

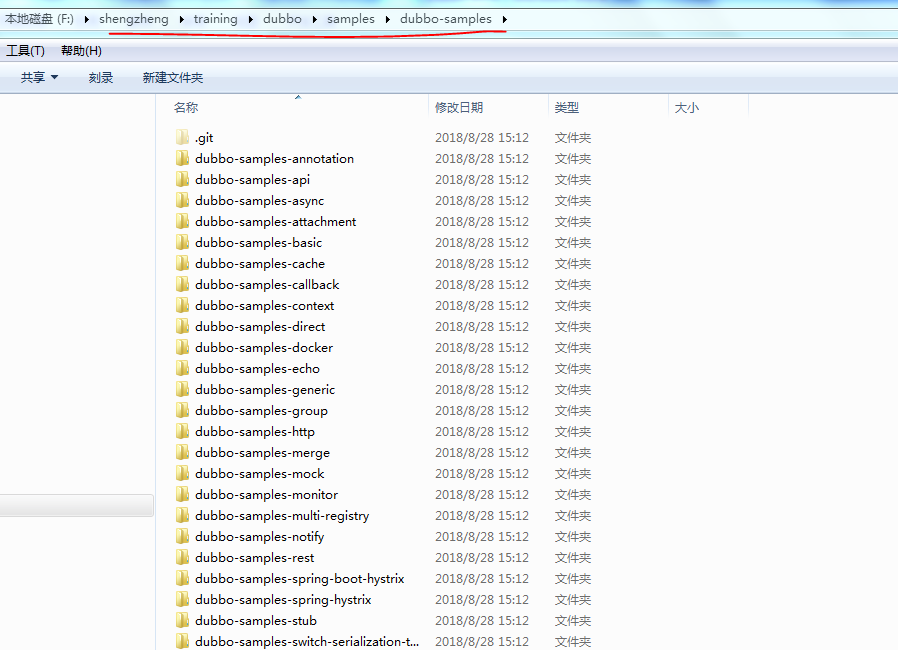
云主机操作系统：



# 1 dubbo-samples-rest

https://github.com/dubbo/dubbo-samples/tree/master/dubbo-samples-rest

$git clone <https://github.com/dubbo/dubbo-samples>



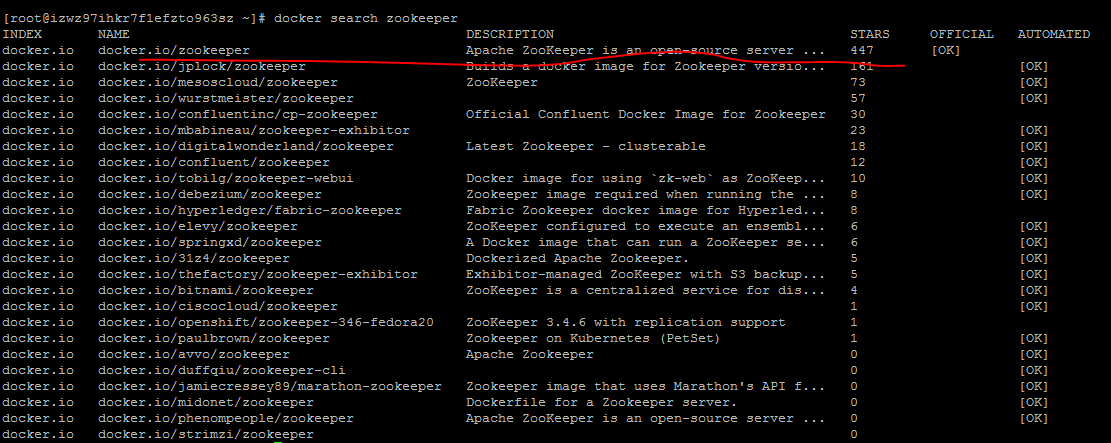
$export MAVEN\_HOME=F:/software/apache-maven-3.5.2

$MAVEN\_HOME/bin/mvn package -X -DskipTests

# 2 zookeeper Docker安装

Docker 中 搜索zookeeper

$docker search zookeeper



$docker pull zookeeper

将下载的文件保存在本地镜像：

$docker save -o /usr/home/softwares/images/zookeeper-image.tar

docker.io/zookeeper



将zookeeper-image.tar载入镜像

docker load -i /usr/home/softwares/images/zookeeper-image.tar

启动zookeeper

$docker run --name zookeeper-1 --restart always -d docker.io/zookeeper

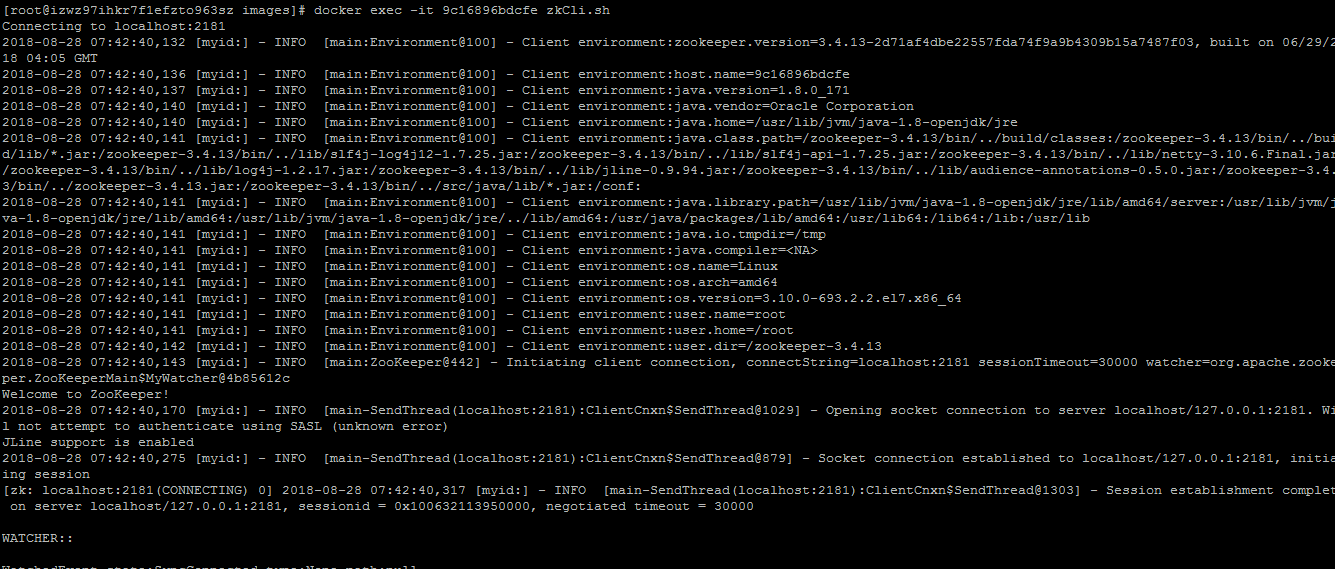
查看正在运行的容器：

$docker ps



启动zookeeper client

$ docker exec -it 9c16896bdcfe zkCli.sh



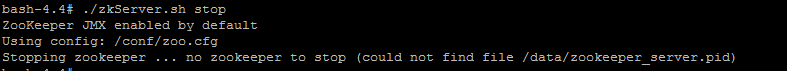
找到安装路径：

$docker exec -it 9c16896bdcfe /bin/bash

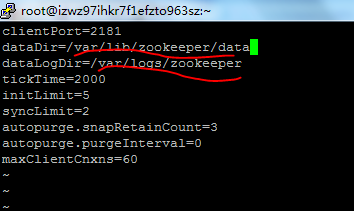
$pwd

停止服务：

$/zookeeper-3.4.13/bin/zkCli.sh stop



需要配置：/conf/zoo.cfg



$mkdir -p /var/lib/zookeeper/data

$mkdir -p /var/logs/zookeeper

$touch /var/lib/zookeeper/data/zookeeper\_server.pid

启动服务端：

$./zkCli.sh -server 127.0.0.1:2181

如果集群，配置文件修改如下：

tickTime=2000

initLimit=10

syncLimit=5

dataDir=/home/dubbo/zookeeper-3.3.3/data

clientPort=2181

server.1=10.20.153.10:2555:3555

server.2=10.20.153.11:2555:3555

使用：

dubbo.registry.address=zookeeper://10.20.153.10:2181?backup=10.20.153.11:2181

或者：

<dubbo:registry protocol="zookeeper" address="10.20.153.10:2181,10.20.153.11:2181" />

# 3 zookeeper官方教程

http://zookeeper.apache.org/doc/current/index.html

## 3.1 什么是zookeeper?

分布应用的分布式协调服务。

## 3.2 zookeeper 特点

* 支持副本集：

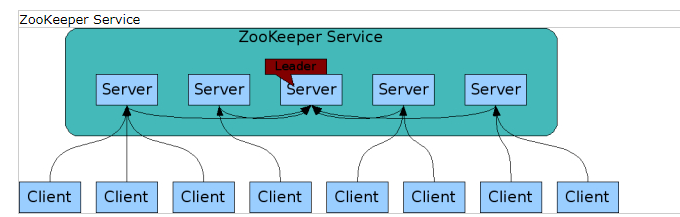


图3.1 zookeeper 副本集

* 组成ZooKeeper服务的服务器必须相互之间能够识别。

维护一个内存的镜像状态，持久存储事务日志和快照。只要主节点在，ZooKeeper服务就可用。

* 客户端连接到单个ZooKeeper服务，维持TCP连接，发送请求，获取响应和观测事件、发送心跳。如果TCP连接终端。客户端将连接到不同的服务节点。
* ZooKeeper有次序

ZooKeeper通过一个数字标记彼此的更新，这个数字反映了所有ZooKeeper的事务。顺次的操作可以使用顺序去实现高阶的抽象，比如：同步化的原型。

* 快速

ZooKeeper 在“读领域”负载尤其的快，ZooKeeper应用运行在成千上万的机器上，在读比写更普遍的场合性能发挥到极致， 如：读写比例10:1左右。

* Data 模型和分层命名空间

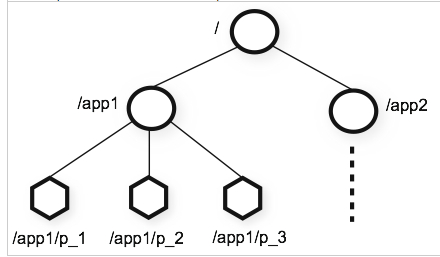


图3.2 zookeeper 分层命名空间

* 节点和短暂节点（ephemeral nodes）

这儿以term Znodes来澄清节点的概念。

Znodes维持一个stat 结构， 该结构包含数据变化的版本，ACL（访问控制列表）变化和时间戳， 以允许缓存校验和协调更新。每当znode数据变化，版本数也增加。例如：任何时候客户端取回的数据的同时也取回了数据的版本。

在命名空间存储在每个znode自动被读写，读获取所有与znode相关的数据字节而写则取代所有的数据。

ZooKeeper也有瞬时节点（ephemeral nodes）的概念，只要创建znode的会话是活跃的，这些瞬时节点都将存在。当会话终止，瞬时节点将删除。

* 条件更新和观察

客户端可以在znodes上设置观察。当客户端收到一个数据包说“znode已经变化了”，观察将触发。当客户端和任意Zoo Keeper 服务断开，客户端将收到位置提醒。

* Zookeeper的保障

Sequential Consistency ：Updates from a client will be applied in the order that they were sent.

Atomicity ：Updates either succeed or fail. No partial results.

Single System Image ：A client will see the same view of the service regardless of the server that it connects to.

Reliability ：Once an update has been applied, it will persist from that time forward until a client overwrites the update.

Timeliness - The clients view of the system is guaranteed to be up-to-date within a certain time bound.

## 3.2 Zookeeper Standardalone 操作

http://zookeeper.apache.org/doc/current/zookeeperStarted.html#sc\_InstallingSingleMode

### 3.2.1 /zookeeper-3.4.13/conf/zoo.cfg

tickTime=2000

dataDir=/var/lib/zookeeper

clientPort=2181

### 3.2.2 非Docker容器启动

$ bin/zkCli.sh -server 127.0.0.1:2181

https://blog.csdn.net/yuanlaijike/article/details/79654183#%E4%BA%8C%E6%90%AD%E5%BB%BAzookeeper

### 3.2.3 windows zookeeper

下载：zookeeper-3.4.10.tar.gz，解压：

启动服务端：

set JAVA\_HOME=F:/software/JDK

set ZOOKEEPER\_HOME=F:/software/zookeeper-3.4.10/zookeeper-3.4.10

%ZOOKEEPER\_HOME%/bin/zkServer start

# 4 安装dubbo-admin.war

## 4.1 download source

<https://github.com/apache/incubator-dubbo/releases>

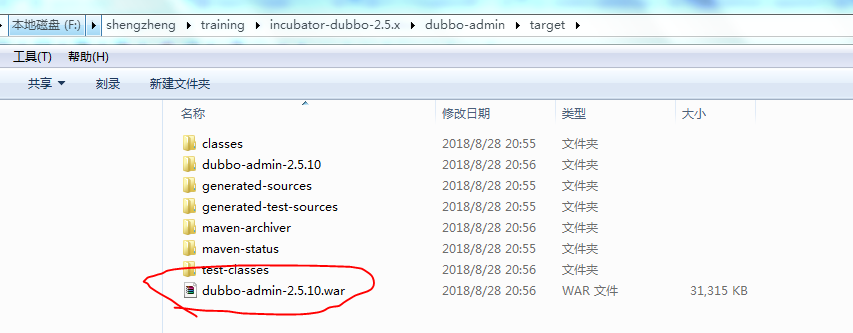
## 4.2 maven compile

$export MAVEN\_HOME=F:/software/apache-maven-3.5.2

$export JAVA\_HOME=F:/software/JDK

$MAVEN\_HOME/bin/mvn package -DskipTests -X

编译完后：产物 dubbo-admin-2.5.10.war



## 4.2 Docker 启动Tomcat

$docker run --name tomcat-1 --restart always -d -p 8081:8080 docker.io/tomcat

## 4.3 upload to cloud

上传文件到云，比如：

$docker cp /root/dubbo-admin-2.5.10.war \

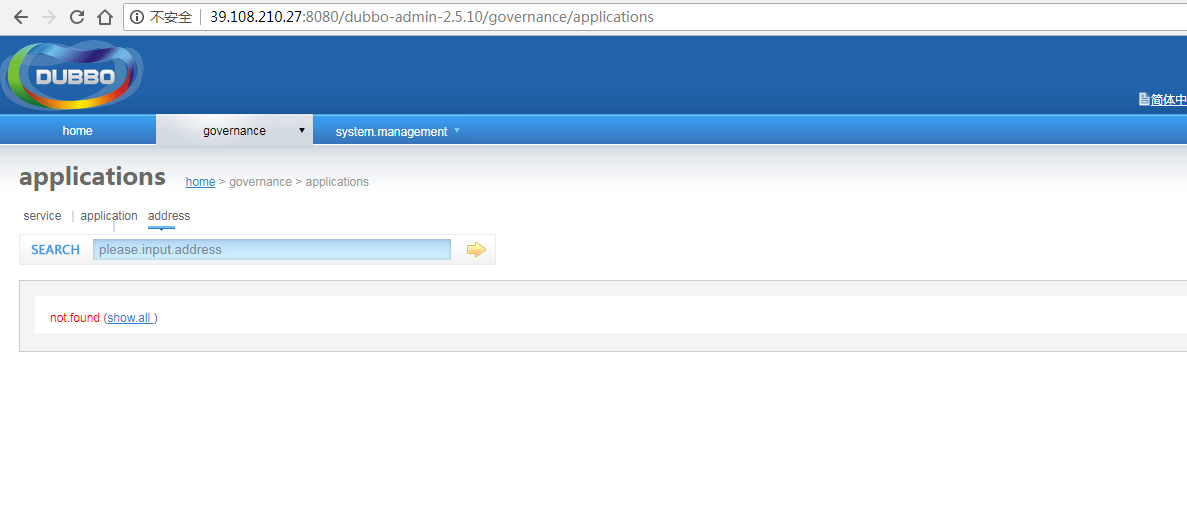
tomcat-1:/usr/local/tomcat/webapps/

$docker run --name tomcat-1 --restart always -d -p 8081:8080 docker.io/tomcat

* 以挂载的形式启动Tomcat

$docker run -d -v /root/dubbo-admin-2.5.10.war:/usr/local/tomcat/webapps/dubbo-admin-2.5.10.war -p 8080:8080 docker.io/tomcat

登陆后台：root/root



如果通过

$netstat -atnp|grep 8080杀掉Tomcat，发现很快又重启了。

# 5 API 调试

<https://www.jianshu.com/p/70151fc0ef5d>

# 6 .YAML 标记语言

http://www.ruanyifeng.com/blog/2016/07/yaml.html?f=tt

# 7 Dockerfile 创建镜像

https://www.cnblogs.com/jie-fang/p/7927643.html

指令 说明

**FROM** 指定所创建镜像的基础镜像

**MAINTAINER** 指定维护者信息

**RUN** 运行命令

**CMD** 指定启动容器时默认执行的命令

**LABEL** 指定生成镜像的元数据标签信息

**EXPOSE** 声明镜像内服务所监听的端口

**ENV** 指定环境变量

**ADD** 赋值指定的<src>路径下的内容到容器中的<dest>路径下，<src>可以为URL；如果为tar文件，会自动解压到<dest>路径下

**COPY** 赋值本地主机的<scr>路径下的内容到容器中的<dest>路径下；一般情况下推荐使用COPY而不是ADD

**ENTRYPOINT** 指定镜像的默认入口

**VOLUME** 创建数据挂载点

**USER** 指定运行容器时的用户名或UID

**WORKDIR** 配置工作目录

**ARG** 指定镜像内使用的参数(例如版本号信息等)

**ONBUILD** 配置当前所创建的镜像作为其他镜像的基础镜像时，所执行的创建操作的命令

**STOPSIGNAL** 容器退出的信号

**HEALTHCHECK** 如何进行健康检查

**SHELL** 指定使用SHELL时的默认SHELL类型

# 8 基于源码安装Dubbo

cd F:\shengzheng\training\incubator-dubbo-dubbo-2.6.3

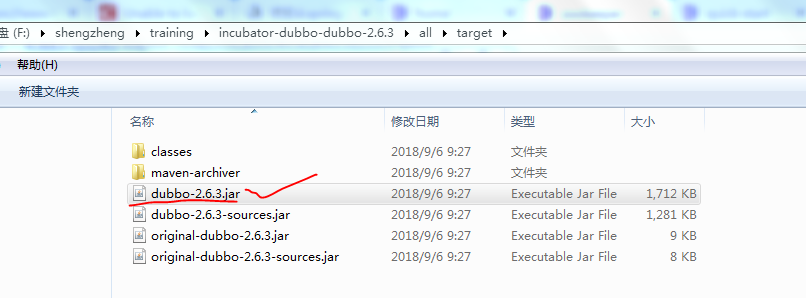
set JAVA\_HOME=F:\software\JDK

set MAVEN\_HOME=F:\software\apache-maven-3.5.2

set MAVEN\_OPTS=-Xmx1024m -XX:MaxPermSize=512m

%MAVEN\_HOME%/bin/mvn install -DskipTests -X

成功安装：



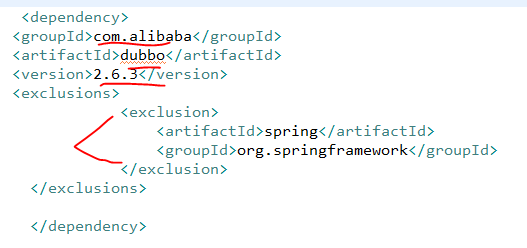
# 9 践行官方实例

## 9.1 使用多路广播中心发现服务。

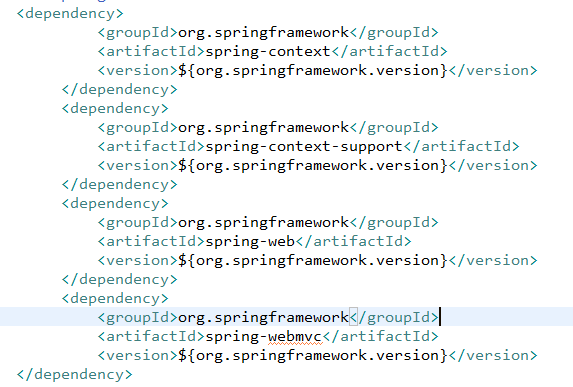
不使用zookeeper完成服务注册，使用多路广播中心。

## 9.2 工程依赖

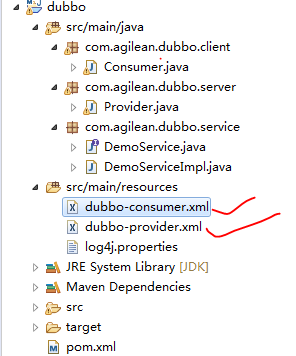
工程中引入：dubbo包，章节8中基于源码安装的jar。必须去除dubbo 自带的老的spring 依赖。



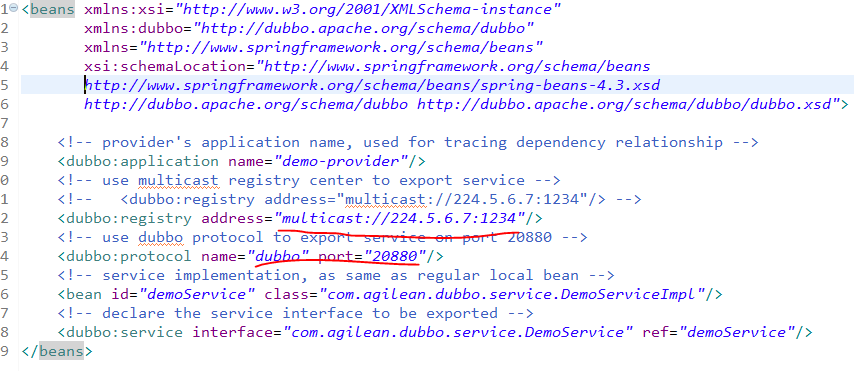
导入:spring4 依赖。使用4.0.8.RELEASE版本。

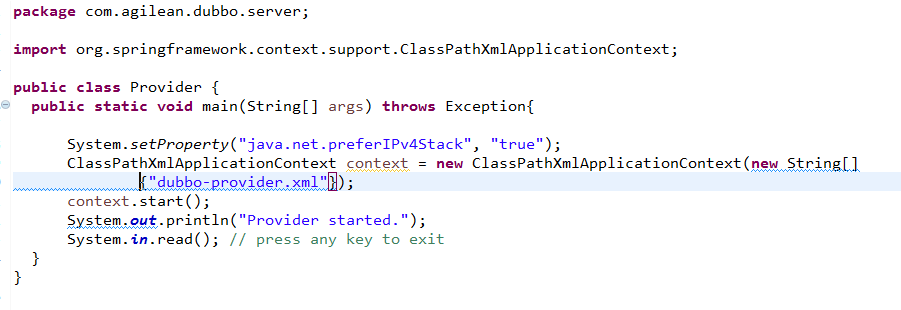


## 9.3 工程结构

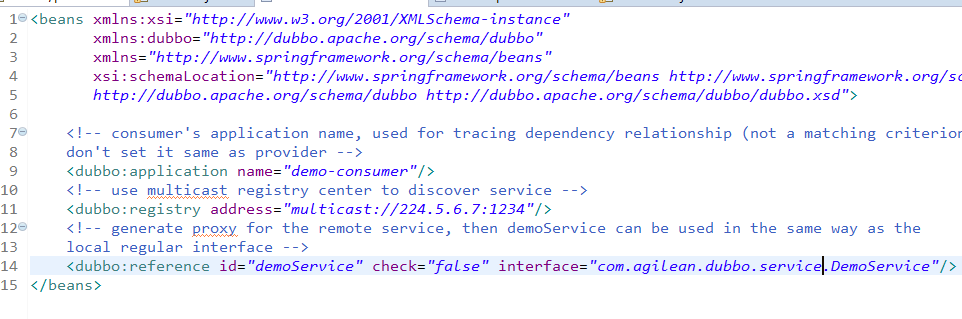


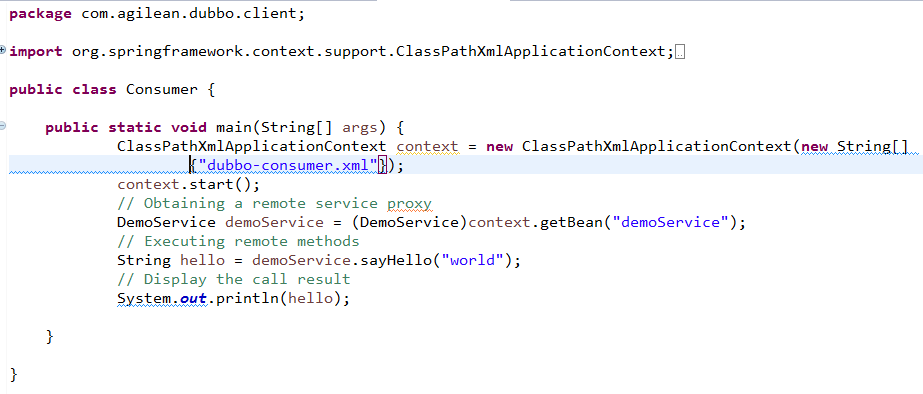
### 9.3.1 服务提供者配置及启动





### 9.3.1 消费者配置及调用服务





# 10 dubbo & zookeeper

dubbo支持多种类型的注册中心：

* Multicast注册中心
* Zookeeper注册中心
* Redis注册中心
* Simple注册中心