**快速掌握Dubbo常规应用**

**概要：**

1. Dubbo 快速入门
2. *Dubbo的作用简要说明、快速演示Dubbo 调用示例*
3. Dubbo 架构与基本角色说明
4. 架构简要讲解、架构中基本角色说明
5. Dubbo基本应用与配置说明
6. 服务提供者常规配置、服务消费者常规配置

Dubbo 快速入门

**Dubbo是什么?他的核心功能是什么？**

dubbo 阿里开源的一个SOA服务治理框架，从目前来看把它称作是一个RCP远程调用框架更为贴切。单从RPC框架来说，功能较完善，支持多种传输和序列化方案。所以想必大家已经知道他的核心功能了：就是远程调用。



**Provider（服务提供者） 应用程序**

**Consumer 应用程序**

**快速演示Dubbo的远程调用**

在pom.xml 当中引入Dubbo 依懒

编写服务接口类

编写服务实现类

编写服务提供者 xml 配置

编写服务消费者 xml 配置

启动服务提供者

编写并执行远程服务调用

**提供者配置----------------------------------**

*<?*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">

<dubbo:application name="demo-provider"/>

<dubbo:registry address="multicast://224.5.6.7:1234"/>

<dubbo:protocol name="dubbo" port="20880"/>

<dubbo:service interface="com.tuling.teach.service.DemoService" ref="demoService"/>

<bean id="demoService" class="com.tuling.teach.service.DemoServiceImpl"/>

</beans>

**消费者配置---------------------------------------**

*<?***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"**>

<**dubbo:application name="demo-consumer"**/>

<**dubbo:registry address="multicast://224.5.6.7:1234"**/>

<**dubbo:reference id="demoService" interface="com.tuling.teach.service.DemoService"** />

</**beans**>

**总结：**

为什么需这些配置?消费者是如何调用到服务提供者的呢？

## **Dubbo 架构与基本角色说明**

**dubbo架构简要讲解**

架构图



**流程说明：**

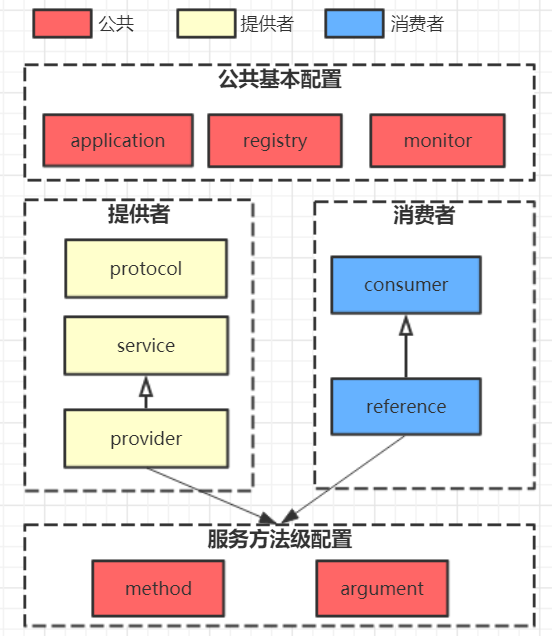
1. Provider(提供者)绑定指定端口并启动服务
2. 指供者连接注册中心，并发本机IP、端口、应用信息和提供服务信息发送至注册中心存储
3. Consumer(消费者），连接注册中心 ，并发送应用信息、所求服务信息至注册中心
4. 注册中心根据 消费 者所求服务信息匹配对应的提供者列表发送至Consumer 应用缓存。
5. Consumer 在发起远程调用时基于缓存的消费者列表择其一发起调用。
6. Provider 状态变更会实时通知注册中心、在由注册中心实时推送至Consumer

**这么设计的意义：**

1. Consumer 与Provider 解偶，双方都可以横向增减节点数。
2. 注册中心对本身可做对等集群，可动态增减节点，并且任意一台宕掉后，将自动切换到另一台
3. 去中心化，双方不直接依懒注册中心，即使注册中心全部宕机短时间内也不会影响服务的调用
4. 服务提供者无状态，任意一台宕掉后，不影响使用

Dubbo常规配置说明

**配置分布图：**



**配置说明：**

|  |  |  |
| --- | --- | --- |
| 标签 | 说明 | 作用 |

**一般配置：**

提供端：---------------------------

<**dubbo:application name="demo-provider"**/><**dubbo:registry protocol="redis" address="192.168.0.147:6379" check="true"**/>

<**dubbo:protocol name="dubbo" port="20880"**/>

<**dubbo:provider group="tuling.dubbo.demo"**

**threadpool="fixed"**

**threads="500"**

**timeout="5000"**

**retries="2"**

/>

<**dubbo:service interface="com.tuling.teach.service.DemoService"**

**timeout="5000"**

**retries="1"**

**version="3.0.0"**

**ref="demoService"**/>

<**bean id="demoService" class="com.tuling.teach.service.DemoServiceImpl"**/>

消费端示例：--------------------

<dubbo:application name="demo-consumer"/>

<dubbo:registry protocol="redis" address="192.168.0.147:6379" check="true"/>

<dubbo:consumer timeout="5000" retries="2"

group="tuling.dubbo.demo"

version="1.0.0"/>

<dubbo:reference

timeout="3000" retries="1"

id="demoService"

version="\*"

interface="com.tuling.teach.service.DemoService"/>

**定位属性:**

定位属性是指消费者匹配提供者的全部标识，总共有如下属性

interface: 接口名称

group**:** 接口分组

version**:** 接口版本

**一般常用属性说明：**

|  |  |  |  |
| --- | --- | --- | --- |
| 标签 | 属性 | 描述 | 必填 |
| application | name | 应用名称 | 是 |
| registry | protocol | 注册中心协议 | 是 |
|  | address | 注册中心地址 | 是 |
| provider | threadpool | 线程模型 fixed/cache | 否 |
|  | threads |  | 否 |
| service | interface |  | 是 |
|  | ref |  | 是 |
|  | group |  | 否 |
|  | version |  | 否 |
|  | timeout | 重试次数 | 否 |
|  | retries | 重试次数 | 否 |
| consumer | timeout |  | 是 |
|  | retries |  | 是 |
| reference | id |  | 否 |
|  | interface |  | 否 |
|  | version |  | 否 |
|  | check |  | 否 |
|  | retries | 重试次数 | 否 |

在提供者配置 在消费端来使用 的属性

timeout

retries

其它属性参考： <http://dubbo.io/books/dubbo-user-book/references/xml/introduction.html>

**属性覆盖：**

