链滴

Dubbo 架构简述

作者：eddy

原文链接：https://ld246.com/article/1510910287499

来源网站：链滴

- 原文地址

全局入口 spring schema handler

spring handler：com.alibaba.dubbo.config.spring.schema.DubboNamespaceHandler，定义了不元素的解析器与映射的类。比如**dubbo:reference**对应**ReferenceBean**，**dubbo:service**对应**ServiceBean**。

provider侧

- 读取spring配置，dubbo会将spring xml中的 **dubbo:service**元素通过预先定义好的解析器封装一个**ServiceBean**，这是一个service的入口。
- **ServiceBean**实现了spring的**ApplicationListener**接口，用来监听spring上下文事件。Dubbo响了**ContextRefreshedEvent**事件，在上下文刷新之后执行**导出provider**动作。
- 执行导出操作，需要两个信息：
  1. 通信协议信息，用来指定consumer和provider间的通信方式以及信息。
  2. 注册中心信息，用来往对应的注册中心注册provider url。
- com.alibaba.dubbo.config.ServiceConfig#doExportUrls生成**registry URL**以及**provider URL**，并将**rovider URL**作为**registry URL**的export参数绑定。根据**registry URL**的协议**registry://拿到对应的egistryProtocol**并调用其**export**方法。
- com.alibaba.dubbo.registry.integration.RegistryProtocol#export主要完成两件事情：
  1. 作为provider，使用定义好的通信协议来暴露服务，如dubbo，http，rmi等。
     - 从 **registry URL**的export参数来获取**provider URL**。
     - 通过 **provider URL**的协议来获取其对应的协议处理类并调用**export**方法。
  2. 使用 **registry URL**中的registry参数作为注册中心协议替换原来的**registry://**协议
     - 通过新的 **registry URL**的协议头，拿到相应的com.alibaba.dubbo.registry.Registry实例并用其com.alibaba.dubbo.registry.RegistryService#register方法，比如向zookeeper新建临时节点。
- 以上，就是provider侧的执行流程简述。

consumer侧

- 同样的，dubbo会将 **dubbo:reference**封装成一个**ReferenceBean**，但与provider不同，consmer实现了org.springframework.beans.factory.FactoryBean接口，初始化的流程推迟到了调用org.springframework.beans.factory.FactoryBean#getObject时。
- 生成 **registry URL**，暂时指定协议为**registry://**，并将实际协议作为**registry**参数的值绑定在UR上。
- 通过 **registry URL**的协议**registry://**，得到相应**RegistryProtocol**，然后调用其com.alibaba.dubbo.registry.integration.RegistryProtocol#refer方法来获取provider引用。
- 更新 **registry URL**协议。
- 通过 **registry URL**获取相应的com.alibaba.dubbo.registry.Registry。
- 使用 **registry URL**和接口类型来生成相应的com.alibaba.dubbo.registry.integration.RegistryDirec

ory。

- RegistryDirectory通过 **registry URL**和接口类型来命名。
- RegistryDirectory用来封装所有的invoker地址信息以及路由信息等。
- 负责通过 **provider URL**中的协议信息获取相应的链接协议处理类并调用**refer()方法以获取nvoker**对象。

- 生成订阅URL，协议为 **consumer://**，通过**registry URL**获取相应的注册中心，向注册中心注册**consumer URL**。
- 调用对应的注册中心的subscribe方法， com.alibaba.dubbo.registry.RegistryService#subscribe。
- 最终，会将 **provider URL**保存到相应的RegistryDirectory中com.alibaba.dubbo.registry.integration.RegistryDirectory#notify。可以查看com.alibaba.dubbo.registry.zookeeper.ZookeeperRegistry#doSubscribe。

请求调用

- 默认使用JDK动态代理模式来获取实际执行类。
- 实际执行对象则是 com.alibaba.dubbo.registry.integration.RegistryProtocol#refer返回的**Invoker**象。
- 最终会执行 com.alibaba.dubbo.rpc.cluster.support.AbstractClusterInvoker#invoke方法。
- 从RegistryDirectory中获取 **Invoker List**。
- 通过负载均衡从 **Invoker List**中获取一个Invoker。
- 拿到对应的链接协议 **Invoker**实例，执行远程调用。
- 注意，consumer实例是由 com.alibaba.dubbo.registry.integration.RegistryProtocol#refer生成的而用来执行远程链接处理远程协议的**refer()**方法则是由RegistryDirectory负责调用的。

以上只是dubbo架构很小的一部分，更多的信息大家可以通阅读dubbo源码获取 dubbo源码地址。