

# dubbo服务相关面试题

## 题目标签

学习时常：20分钟

题目难度：初级

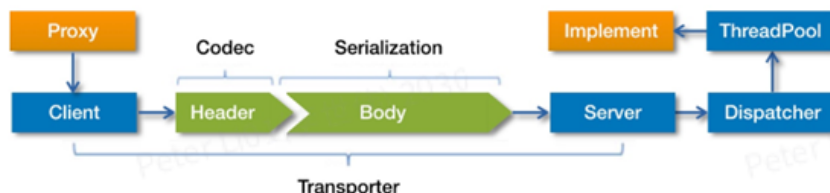
知识点标签：Dubbo、代理、Netty、Zookeeper

## 题目描述

请介绍一下Dubbo服务消费者调用服务提供者的过程？

## 题目解决

### 1. 整体流程图



通过上面的流程图可以看到，服务消费者通过代理对象 Proxy 发起远程调用，接着通过网络客户端 Client 将编码后的请求发送给服务提供方的网络层上，也就是 Server。Server 在收到请求后，首先要做的事情是对数据包进行解码。然后将解码后的请求发送至分发器 Dispatcher，再由分发器将请求派发到指定的线程池上，最后由线程池调用具体的服务。这就是一个远程调用请求的发送与接收过程。

### 2. 执行流程

#### 2.1 生成动态代理类

当调用服务方法的时候，会被InvokerInvocationHandler类的invoke方法拦截

```
public Object invoke(Object proxy, Method method, Object[] args) throws Throwable{
```

2.2 将封装参数到 RpcInvocation 中执行后续的调用

```
return invoker.invoke(new RpcInvocation(method, args)).recreate();
```

2.3 没有服务降级逻辑

没有服务降级逻辑，即无 mock 逻辑，直接调用其他 Invoker 对象的 invoke 方法，比如

```
FailoverClusterInvoker result = **this**.invoker.invoke(invocation);
```

2.4 抽象方法，由子类实现

```
return doInvoke(invocation);
```

2.5 发送请求

得到一个 ResponseFuture 实例，并调用该实例的 get 方法进行等待

```
return (Result) currentClient.request(inv, timeout).get();
```

2.6 写数据

```
(1) channel.request(request, timeout);(2) channel.send(req);(3) ChannelFuture future =  
channel.writeAndFlush(message);(4) pipeline.writeAndFlush(msg);(5) ExchangeCodec类对  
请求数据进行编码和序列化 (6) next.invokeWrite(m, promise);
```

---

## 题目描述

请介绍一下Dubbo服务提供者接收消费者请求的过程？

---

## 题目解决

默认情况下 Dubbo 使用 Netty 作为底层的通信框架。Netty 检测到有数据入站后，首先会通过解码器对数据进行解码，并将解码后的数据传递给下一个入站处理器的指定方法。

## 执行流程

(1) 执行ExchangeCodec类解码方法

```
public Object decode(Channel channel, ChannelBuffer buffer) throws IOException
```

(2) 解码后得到一个Request对象

(3) 解码器将数据包解析成 Request 对象后，NettyServerHandler 的 channelRead 方法紧接着会收到这个对象，并将这个对象继续向下传递。这期间该对象会被依次传递给 AbstractPeer、MultiMessageHandler、HeartbeatHandler

以及 AllChannelHandler。最后由 AllChannelHandler 将该对象封装到 Runnable 实现类对象中，并将 Runnable 放入线程池中执行后续的调用逻辑。

(4) 接下来会调用 ChannelEventRunnable，ChannelEventRunnable 将会是服务调用过程的新起点

// 将 channel 和 message 传给 ChannelHandler 对象，进行后续的调用// handler是 DecodeHandler类型对象handler.receive(channel, message);

(5) 执行 HeaderExchangeHandler 类方法

public void received(Channel channel, Object message)

在该方法内部执行 handleRequest 逻辑

handleRequest(exchangeChannel, request)

(6) 执行 DubboProtocol 类的 invoke 方法

Result result = invoker.invoke(inv);

在该方法内部，调用我们的目标实现类内容

return wrapper.invokeMethod(proxy, methodName, parameterTypes, arguments);

至此，调用到了目标实现类的方法

handler.receive(channel, message);

---

## 题目扩展

由于 dubbo 天然依赖于 spring 框架，能否自己手动编写一个 mini 版 Dubbo 框架，并且不依赖 Spring 框架且基于 netty 或者 socket 实现网络通信

---

## 题目扩展思路

1.1 服务注册到注册中心

1.2 编写 RPC 服务类

(1) 绑定服务名称和服务对象

(2) 发布服务到注册中心，并且等待客户端调用

### 1.3 编写消费者

(1) 从注册中心找到服务

(2) 创建XxxService接口的代理类对象

(3) 使用代理对象远程调用方法

### 1.4 编写服务发现

(1) 连接zookeeper

(2) 发现服务，获取zk节点，从而读取zk上面的数据

### 1.5 编写代理服务对象

### 1.6 编写Socket客户端发起远程调用

(1) 创建socket客户端对象

(2) 远程调用服务提供者