1、面试题

如何自己设计一个类似dubbo的rpc框架？

2、面试官心里分析

说实话，就这问题，其实就跟问你，如何自己设计一个MQ，一样的道理，就考两个：

（1）你有没有对某个rpc框架原理有非常深入的理解

（2）你能不能从整体上来思考一下，如何设计一个rpc框架，考考你的系统设计能力

3、面试题剖析

其实一般问到你这问题，你起码不能认怂，因为既然咱们这个课程是短期的面试突击训练课程，那我不可能给你深入讲解什么kafka源码剖析，dubbo源码剖析，何况我就算讲了，你要真的消化理解和吸收，起码个把月以后了。

所以我给大家一个建议，遇到这类问题，起码从你了解的类似框架的原理入手，自己说说参照dubbo的原理，你来设计一下，举个例子，dubbo不是有那么多分层么？而且每个分层是干啥的，你大概是不是知道？那就按照这个思路大致说一下吧，起码你不能懵逼，要比那些上来就懵，啥也说不出来的人要好一些

举个例子，我给大家说个最简单的回答思路：

（1）上来你的服务就得去注册中心注册吧，你是不是得有个注册中心，保留各个服务的信心，可以用zookeeper来做，对吧

（2）然后你的消费者需要去注册中心拿对应的服务信息吧，对吧，而且每个服务可能会存在于多台机器上

（3）接着你就该发起一次请求了，咋发起？蒙圈了是吧。当然是基于动态代理了，你面向接口获取到一个动态代理，这个动态代理就是接口在本地的一个代理，然后这个代理会找到服务对应的机器地址

（4）然后找哪个机器发送请求？那肯定得有个负载均衡算法了，比如最简单的可以随机轮询是不是

（5）接着找到一台机器，就可以跟他发送请求了，第一个问题咋发送？你可以说用netty了，nio方式；第二个问题发送啥格式数据？你可以说用hessian序列化协议了，或者是别的，对吧。然后请求过去了。。

（6）服务器那边一样的，需要针对你自己的服务生成一个动态代理，监听某个网络端口了，然后代理你本地的服务代码。接收到请求的时候，就调用对应的服务代码，对吧。

这就是一个最最基本的rpc框架的思路，先不说你有多牛逼的技术功底，哪怕这个最简单的思路你先给出来行不行？好，突击课程，那就到这儿结束了，我这课程定位是帮你快速梳理一遍，扫清盲点，不是打通你任督二脉，给你九阳神功的。