

前后端分离是如何做的

在前后端分离架构中，后端只需要负责按照约定的数据格式向前端提供可调用的 API 服务即可。前后端之间通过 HTTP 请求进行交互，前端获取到数据后，进行页面的组装和渲染，最终返回给浏览器。	后端	前端	
服务器	浏览器		
JAVA	NodeJS	JS + HTML + CSS	
服务层提供数据接口维持数据稳定封装业务逻辑	跑在服务器上的 JS 转发数据，串接服务路由设计，控制逻辑渲染页面，体验优化更多的可能	跑在浏览器上的 JSCSS、JS 加载与运行 DOM 操作任何的前端框架与工具共用模版、路由	

微服务哪些框架

Dubbo，是阿里巴巴服务化治理的核心框架，并被广泛应用于阿里巴巴集团的各成员站点。阿里巴巴近几年对开源社区的贡献不论在国内还是国外都是引人注目的，比如：

JStorm 捐赠给 Apache 并加入 Apache 基金会等，为中国互联网人争足了面子，使得阿里巴巴在国人眼里已经从电商升级为一家科技公司了。

Spring Cloud，从命名我们就可以知道，它是 Spring Source 的产物，Spring 社区的强大背书可以说是 Java 企业界最有影响力的组织了，除了 Spring Source 之外，还有

Pivotal 和 Netflix 是其强大的后盾与技术输出。其中 Netflix 开源的整套微服务架构套件是 Spring Cloud 的核心。

说说 RPC 的实现原理

首先需要处理网络连接通讯的模块，负责连接建立、管理和消息的传输。其次需要有编解码的模块，因为网络通讯都是传输的字节码，需要将我们使用的对象序列化和反序列化。剩下的就是客户端和服务端的部分，服务器端暴露要开放的服务接口，客户调用服务接口的一个代理实现，这个代理实现负责收集数据、编码并传输给服务器然后等待结果返回。

说说 Dubbo 的实现原理

dubbo 作为 rpc 框架，实现的效果就是调用远程的方法就像在本地调用一样。如何做到呢？就是本地有对远程方法的描述，包括方法名、参数、返回值，在 dubbo 中是远程和本地使用同样的接口；然后呢，要有对网络通信的封装，要对调用方来说通信细节是完全不可见的，网络通信要做的就是将调用方法的属性通过一定的协议（简单来说就是消息格式）传递到服务端；服务端按照协议解析出调用的信息；执行相应的方法；在将方法的返回值通过协议传递给客户端；客户端再解析；在调用方式上又可以分为同步调用和异步调用；简单来说基本就这个过程

- 怎么理解 RESTful
<http://www.cnblogs.com/artech/p/3506553.html>
- 说说如何设计一个好的 API
<https://juejin.im/entry/59b8d34c6fb9a00a4455dd04>
- 如何理解 RESTful API 的幂等性
http://blog.720ui.com/2016/restful_idempotent/
- 如何保证接口的幂等性
<http://www.spring4all.com/article/914>
- 说说 CAP 定理、BASE 理论
<http://my.oschina.net/foodon/blog/372703>
- 怎么考虑数据一致性问题
<https://opentalk.upyun.com/310.html>
- 说说最终一致性的实现方案
<http://www.cnblogs.com/soundcode/p/5590710.html>
- 你怎么看待微服务
<http://dockone.io/article/394>
- 微服务与 SOA 的区别
<http://dockone.io/article/2399>
- 如何拆分服务
<http://dockone.io/article/2516>
- 微服务如何进行数据库管理
<http://www.uml.org.cn/wfw/201705271.asp>
- 如何应对微服务的链式调用异常

http://blog.720ui.com/2017/msa_design/?utm_source=tuicool&utm_medium=referral

- 对于快速追踪与定位问题
依赖日志

- 微服务的安全

<http://dockone.io/article/1507>