

# 「2022面经」：2年前端拿下字节阿里offer总结

## 前言

本文主要给大家带来今年 2022 年，互联网寒冬下，我的一些面试经历和经验，希望这篇文章能给正在找工作的同学们一点参考和帮助。

先说一下我的个人情况：本科毕业 工作经验 2 年，2020年7月 入职上一家公司，技术栈为 Vue，在职期间做过 前端监控SDK、监控告警平台、在线协作文档&表格 等，参与过 技术选型调研、架构设计 等；

投了简历后面了 字节、阿里，还有一些 中小公司、非互联网公司，最后拿下 字节、阿里等 offer，入职了 阿里；

- 总的来说今年确实寒冷，到处都在缩招，有的公司连面试机会都不给；

本篇文章分为两部分：面试准备 & 面经

面经一共三篇：CVTE、字节、阿里；

CVTE 面经是让大家了解一下 非BAT公司，在今年 2022 环境下所问的问题；

只想看 字节、阿里 面经的，直接跳到最后即可~

## 关于面试准备

### 简历准备

我的简历就是用 Markdown 做了一份，然后包含了以下这些方面：

- 个人资料：姓名、电话、邮箱、大学（本科or硕士/博士）、博客地址（可选）
- 专业技能：列举你的亮点技能即可！
- 工作经历：写清公司名称、工作时间；如果经历了多家公司，写一写每家公司负责的事情；

- 项目经验：列举几个你参与最深，最能体现技术深度的项目即可；
- 技术开源：如果你有发起开源项目且有 200 个以上的start，或者参与到一些知名的开源项目建设里就可以写一写；（如果PR只是改了几个错别字就不要写了，笑哭QWQ）

小提示：如果你不是美若天仙，我这里不建议在简历里放照片；（笑哭QWQ）

小提示：个人资料里面，邮箱和电话非常的重要，因为很多公司的远程面试邀请是需要发到你的邮箱的！电话则是跟你约面试的时间还有沟通offer等等；

小提示：写专业技能的时候，请不要将你所有知晓过的技能都罗列出来；这会让人觉得没有重点，列举你跟岗位相关且熟悉的技能，不要给自己去挖坑；

我的简历里专业技能怎么写的呢？我写的很简单：熟悉JavaScript、typescript、Vue.js、Nodejs；没了，尽量不要给自己去挖坑；

小提示：工作经历里面，如果你在上一家公司里面，有比较优秀的绩效或者有晋升经历，可以点一笔，亲身经历：写了后HR面的时候她能直接知道你的绩效优秀，先入印象；

## 基础准备

基础知识（八股文）还是很重要的

这里附上我自己总结的基础知识点文章，希望我的总结凝练能够给同学们一点帮助和启发；文章中如果有写错的，请在对应文章下留言评论，如果有帮助，请三连（点赞评论+收藏）

专栏：[2022寒冬下的前端面试知识点总结：](#)

[「2022」寒冬下我的万字面试准备【浏览器原理&安全】篇](#)

[「2022」寒冬下我的万字面试准备【计算机网络】篇](#)

[「2022」寒冬下我的万字面试准备【JS】篇](#)（加紧编写中）

[「2022」寒冬下我的万字面试准备【CSS】篇](#)（加紧编写中）

## 项目复盘 & 深挖

项目经验很重要，因为一般 2~3 面开始，就侧重在项目深度的挖掘上了；

- **项目动机**：需求、为什么做、项目对公司来说有什么意义；
- **负责角色**：在项目中负责的什么角色，是 **Owner**、**核心开发**、还是只是 **滑滑水**；
- **技术选型**：这个项目的选型设计（如果有技术选型的话）
- **架构设计**：架构设计和分层设计，如果你有 **参与或者是独自架构** 的，可以选择在简历上直接附图（画成横向的）
- **细节实现**：一些具体的细节点如何实现，比如监控SDK就有各个监控数据的细节采集方式；
- **亮点设计**：做这个项目中的一些难点考虑，亮点设计等；
- **项目成果**：表明你这个项目达成了什么成果，最好有数据说话；

最后我给自己总结了个 2W5K 字的复盘..... 见图：

然后这个 2W5K 字的复盘确实最后也帮助我拿到了 字节和阿里的offer；

字数统计

总字数 24,067

总字符数 31,110

@稀土掘金技术社区

## HR面准备

需要提前准备的一些常规问题还是挺多的，包括：

- 自我介绍
- 为什么要离职

- 有成就感的事
- 在原公司的收获
- 业余生活
- 对新公司、新团队的期待
- 遇到的最大挫折，怎么应对
- 对城市的选择

如果是有 **知名产品** 的公司、团队面试，你还需要提前了解一下对应的产品

- 使用 **XX** 的过程中有没有什么觉得不好的地方
- .....

## 关于面试

---

### 简历投递

**简历投递** 一定要找内推！一定要找内推！一定要找内推！

有内推和没内推说实话还是 **有差别的**，同学们如果能找到内推渠道的话一定要找内推进行简历投递，

### 面试过程中

面试过程中提醒大家几个点：

- 简历中先等待面试官将问题说完后，再进行自己回答的阐述； **（尽量不要抢答）**
- 可以先花个 **3~5** 秒的时间，将自己的回答梳理一下再输出；
- 自我介绍一定要 **提前准备**！
- 每次面试前先想好面试结束的反问需要询问的内容；

### 阿里、字节面试感受

这次大厂面了 **阿里 + 字节**，大体说下感受：

- 阿里全程没有问过一个 八股文 问题，所有时间都在深挖 项目 的 细节、 架构设计逻辑 还有 合理性；
- 字节还是常规的，一面以基础为主；然后之后的以 项目深度实现 为主展开问答；

总体感受就是，如果你做的项目很复杂，有足够的深度给面试官挖，那么你就好好的准备你的项目，不然就好好准备基础知识；

准备项目需要做到这个方面的总结： 为什么做、 怎么做、 有什么难点&问题、 业界怎么做 更好；

## CVTE

### 一面

### 自我介绍

上来后面试官也不整聊天活跃一下气氛，直接就让介绍一下你自己（说出你的故事）

小提示：自我介绍尽量保持在 2~3 分钟之间；语速均匀，不要结巴；可以说一下自己非常熟悉的项目， 试着将面试官的注意力往自己熟悉的方向拽；

### 你知道 BFC 么

小提示：先聊一聊 BFC 是什么；

然后再说说触发 BFC 的条件；

再说一说 BFC 的一些作用

BFC 是块级格式化上下文，是一个独立的渲染区域，让处于 BFC 内部的元素与外部的元素相互隔离，使内外元素的定位不会相互影响。

### 触发条件

- position: absolute / fixed：绝对定位
- display: inline-block / table / flex

- `float` 设置除 `none` 以外的值；（只要设置了浮动，当前元素就创建了BFC）
- `overflow != visible`（可为：`hidden`、`auto`、`scroll`）

## 特性和应用

- 阻止 `margin` 重叠：同一个 BFC 下外边距（margin）会发生折叠
- 清除浮动：清除内部浮动(清除浮动的原理是两个 `div` 都位于同一个 BFC 区域之中)
- 自适应两栏布局：左 `float` + 右 BFC，是利用了 BFC 的区域不会与 `float` 的元素区域重叠的机制

## 说说 Flex

**小提示：**先聊一下概念：`Flex` 通常被称为 `flexbox`，是一种弹性盒布局模型，通常用作解决布局问题；

然后再说一说 `Flex` 三个值（`flex-grow`、`flex-shrink`、`flex-basis`）以及 `flex 1`、`flex 0`、`flex auto` 的取值；

然后再说一说使用 `Flex` 布局时需要注意的点；

然后再说一说 `Flex` 在布局上的一些应用；

## 基本概念

- 通常被称为 `flexbox`，是一种一维的布局模型，给予元素提供了空间分布和对齐能力。它由（`Flex Container` 容器/ `Flex item` 项目成员）构成。
- 该布局模型的目的是提供一种更加高效的方式来对容器中的条目进行布局、对齐和分配空间。适用于不同尺寸屏幕中创建可自动扩展和收缩布局，通常可用于水平垂直居中，两栏、三栏布局 等的场景里

### Flex 三个值

- `flex-grow`：项目的放大比例，默认为 `0`，即如果存在剩余空间，也不放大。
  - 如果所有项目的 `flex-grow` 属性相等（或都为1），将等分剩余空间，如果有一个为 `2`，那么它占据的剩余空间将比其他项目多；
- `flex-shrink`：项目的缩小比例，默认为 `1`，即如果空间不足，该项目将缩小
- `flex-basis` 属性定义了分配多余空间之前，项目占据的主轴空间（`main size`）。浏览器根据这个属性，计算主轴是否有多余空间。它的默认值为 `auto`，即项目的本来大小。

- `flex` 属性是 `flex-grow`, `flex-shrink` 和 `flex-basis` 的简写, 默认值为 `0 1 auto`。后两个属性可选。该属性有两个快捷值: `auto (1 1 auto)` 和 `none (0 0 auto)`。
- 如果设置 `flex:1`, 就等于 `flex: 1 1 0`; 设置 `flex:0`; 就等于 `flex: 0 0 0`;

## 子元素都设置 flex 1 宽度是否一样;

**小提示:** 这个问题是对上述 `Flex` 的一个追问

`Flex 1` 相当于 `flex: 1 1 0`; 项目占的主轴空间为 `0`, 所以平分; 但是如果有 `padding` 属性的话, 元素占的地方会被增加; 设置 `margin` 的话, 元素的宽度不变, 但是 `content` 内容的宽度变小;

## Flex 如何实现三栏布局

左右设置 `flex: 0 1 200px`; 中间设置 `flex:1`; 父元素 `flex` 布局。

## Flex 1 的理解

如果设置 `flex:1`, 就等于 `flex: 1 1 0`; 设置 `flex:0`; 就等于 `flex: 0 0 0`;

## 有20个异步请求, 如何保持同时三个的并发

**小提示:** 我这里直接聊的实现思路: 用 `Promise.race` 来实现

## Promise 输出结果题

**小提示:** 原题不是这个, 是一个类似的题, 但是原理都一样;

js 复制代码

```
const p = new Promise((resolve, reject) => {  
  console.log(0)  
  reject()  
  console.log(1)  
  resolve()  
  console.log(2)  
})  
p.then(res => {
```

```
console.log(3)
}).then(res => {
  console.log(4)
}).catch(res => {
  console.log(5)
}).then(res => {
  console.log(6)
}).catch(res => {
  console.log(7)
}).then(res => {
  console.log(8)
})
```

**小提示：**这里主要想考察的是两个点：

- 1.在 `Promise` 中，`resolve` 或者 `reject` 不会阻止 `Promise` 内的其余代码的执行
- 2.并且后面的 `then` 和 `catch`，会链式调用；

结果为： 0 1 2 5 6 8

## 箭头函数和普通函数的区别

常规特性问题

- 函数体内的 `this` 对象，就是定义时所在的对象，而不是使用时所在的对象
- 不可以当作构造函数，也就是说，不可以使用 `new` 命令，否则会抛出一个错误
- 不可以使用 `arguments` 对象，该对象在函数体内不存在。如果要用，可以用 `Rest` 参数代替

## 聊聊 Set Map

**小提示：**我们可以先聊一下 `Set` 和 `Map` 的特点；

然后拓展出 `WeakSet` 和 `WeakMap`，说说区别（使用上的），场景（弱引用）

### set 和 weakSet 区别

[MDN: WeakSet](#)

- `Set` 允许存储任何类型的唯一值（不能重复），无论是原始值或者是对象引用；



- `WeakSet` 成员都是弱引用的 `对象`，会被垃圾回收机制回收，可以用来保存 `DOM` 节点，容易造成内存泄漏；
- `WeakSet` 不可迭代，不能用在 `for-of` 循环中
- `WeakSet` 没有 `size` 属性

## map 和 weakMap 的区别

- `Map` 的 `key` 可以是任意的数据类型（基础类型、对象、函数等）；`weakmap` 的键只能是非 `null` 的对象引用；
- `Map` 的 `key` 是强引用（只要键不释放就会一直占着内存，不会被 `GC`），`weakmap` 的 `key` 是弱引用的对象，所以不会计入垃圾回收引用次数（在没有其他引用存在时垃圾回收能正确进行）；
- `Map` 能轻易转化为数组（`扩展运算符`）；`weakmap` 做不到
- 由于 `key` 随时会被回收，所以 `weakmap` 的 `key` 不可枚举，相应地也就不能获取 `size` 等，它能做的事情也就只有 `has/get/set/delete` 四种操作；`map` 相对比较丰富，`has/get/set/delete` 之外，支持 `entries/size/foreach/keys/values` 等

## Map 和 普通Object的区别

过去通常用 `object` 实现，但是 `obj` 只能用字符串作为 `key`，有很大限制，所以出现 `map`，支持任意类型作为 `key`；

## 防抖节流了解吗

**小提示：**这里聊一聊节流和防抖的概念，还有适用场景分别是什么；

- **防抖**：是指在事件被触发 `n` 秒后再执行回调，如果在这 `n` 秒内事件又被触发，则重新计时。这可以用在键盘输入上，等用户输入完成时自动进行字符串校验等
- **节流**：是指规定一个单位时间，在这个单位时间内，只能有一次触发事件的回调函数执行，如果在同一个单位时间内某事件被触发多次，只有一次能生效。节流可以使用在 `scroll` 函数的事件监听上，通过事件节流来降低事件调用的频率。

## 手写的实现思路

**小提示：**这里没让我直接手撸，而是问思路有一点意外

- 节流触发事件时，设置一个定时器，再次出发事件时定时器存在，不会再执行，直到定时器到时候，清空定时器，执行函数；
- 防抖也是设置一个定时器，但是再次触发事件时，清空定时器重新进行计时；

## 浏览器输入 url 后到展示页面的过程

**小提示：**这个问题可太经典了，但是也可以趁机说出一个很完整的答案；**具体见我的总结；**

[「2022」寒冬下我的万字面试准备【计算机网络】篇：【从输入URL到呈现页面过程】](#)

## 性能优化方面是怎么做的

**小提示：**可以分块进行总结回答；**具体见我的总结；**

- 重定向优化：
  - 避免重定向
- DNS 、 TCP 优化：
  - 预解析；也就是用 `<link rel="preconnect" href="//`sample.com`" crossorigin>`
  - 使用 `preconnect` 时，同时也使用 `dns-prefetch` 作为兼容手段，或者凸显关键资源；
- 请求优化：
  - 减少 HTTP 请求数量：多个压缩成一个；
  - 使用 HTTP2：头部压缩
  - 减少 cookie 大小：http 请求会携带 cookie
- DOM 、 CSS 解析优化：
  - CSS 放在头部
  - 优化 CSS 的 选择器 结构：浏览器解析 CSS 选择器是从右到左；
  - JS 放在尾部，或者标注 `defer` 、 `async` 避免阻塞解析；
  - 精简 DOM 结构
  - 提取 `critical css`：也就是将首屏需要用到的 CSS 放在 Head 里加载，其余的异步加载；
- 渲染优化：
  - 减少 DOM 操作的次数，减少重绘重排
  - 减少 DOM 数量：大数据量分页、虚拟列表
  - 使用 GPU 加速：比如 `transforms` 和 `opacity` 两个属性；

- 要慎用，低端机 GPU 差，占用较多内存卡顿，因此什么时候开启硬件加速，给多少元素开启硬件加速，需要用测试结果说话
- 尽量使用 `requestAnimationFrame` 来实现视觉变化
- 使用 `loading` 图：能较早的有视觉变化
- 三方资源、图片优化
  - 第三方资源、库使用 `CDN`
  - 压缩图片体积，减少图片大小；
  - 图片懒加载：使用 `IntersectionObserver` 或者监听 `scroll` 滚动的距离
  - 图片使用 `webp` 格式：体积会比 `jpg` 小；
  - 去掉图片中的 `metadata`（拍照时的时间地点等）
  - 使用 `preload` 预先加载 `css文件` 或者 `字体文件`、`js文件` 等；从而更不易阻塞页面的初步渲染，进而提升性能
  - 还可以关注一下 `HTTP2服务端推送` 或者 `HTTP103状态码`；
- 其余的性能优化
  - 合理使用 `协商缓存`
  - 善于使用 `事件委托`
  - 避免死循环（死循环导致页面卡死）
  - 慎用闭包、全局状态（导致内存泄漏）
  - 使用 `Web Workers` 处理纯数据，或者与浏览器 `UI` 无关的长时间运行脚本
  - 保持代码设计的优雅性，减少冗余代码；
  - 灵活使用防抖节流；
  - 避免频繁 `JS` 获取 `Layout` 属性值造成的回流；

## SPA应用如何采集路由切换性能

**小提示：**我这里先回答我们目前没有实现这一方面的功能，但是我们做了相应的方案调研，然后再抖出自己的方案；

**方案：**可以利用 `MutationObserver` API，从切换路由时开始计时，直到 `MutationObserver` 监听 `DOM` 稳定时算做切换完毕；这一段耗时就可以认为是路由切换性能；

## 如何监听路由切换

**小提示：**先聊聊路由分为 `Hash` 路由和 `History` 路由；

然后说明 Hash 路由的监听可以通过监听 hashchange ,或者只监听 hashchange

最后再说明 History 路由的监听方式

## [一文摸清前端监控自研实践（二）行为监控：【路由跳转】](#)

### 如何监听所有的页面按钮点击

**小提示：**可以参考我之前写的一篇文章中的做法；

## [一文摸清前端监控自研实践（二）行为监控：【点击事件】](#)

### 如何监控一个人的在线时间

**小提示：**在线时间，可以理解为这个人这个页面的时间，可以通过监听 visibility 事件进行统计；

### 监控数据采集时是否会丢数据

**小提示：**这里回答清楚自己前端监控 SDK 对于监控数据上报失败的重传逻辑就可以了；

### 性能上报上报哪些数据

**小提示：**我回答这个问题，都会先回答：我将性能指标分为 以用户为中心的性能指标 和 以技术为中心的性能指标 ，然后再分开聊；

## [一文摸清前端监控自研实践（一）性能监控：【以用户为中心的性能指标】](#)

### 开发过程中遇到的技术难题

**小提示：**这个问题就大家仁者见仁智者见智看着回答了；

## 二面

## 项目问答

二面项目面就是对着你的简历，根据你写的内容一个一个的问；你写在简历上的，都会成为问的点；

因为没有什么难题，问的都是属于简历上写的什么，就问什么，就不列举了；想看项目问题的，[直接跳转下面的字节和阿里](#)；

## 概率论问题

当时二面问了我一个 [概率学](#) 的问题，大学虽然有学过概率学，但是工作了两年都忘了；当时面试官拿这题折磨了我十分钟，很难顶；不懂这题的意义在哪里；

## 手写笔试

类似于 [岛屿数量](#) 这一题的改装题；

### [200. 岛屿数量](#)

## HR面

### 大致问题：

HR面有一个小插曲：当时HR问我个性层面有什么缺点，我想了想说个性层面没有，我有的时候讨论方案的时候说话语速很急算不算；

然后HR说不算，非得让我想一个，无语了都；

- 看工作机会出于什么考虑
- 是否觉得在现在的公司发展受限
- 薪资情况、期望薪资
- 个人职业规划
- 平常如何学习

- 为什么学习前端
- 原公司挽留你怎么办
- 我们公司吸引你的点在哪里
- 是否有其余的 Offer
- 家庭情况，父母在哪，打算在哪个城市定居
- 在工作中不太希望、喜欢发生什么事情
- 跟组内成员的协作情况如何
- 你是一个什么样的人
- 觉得自己有什么不足，个性层面的缺点有什么
- 同事是否有对你提出什么意见
- 周末怎么安排
- 理想的工作节奏、时间
- 在新环境发展一段时间后感觉不合适自己，你会怎么做

## 字节

---

### 一面技术面

上来后，面试官先让做个 **自我介绍**，然后简单唠了唠，**活跃了下气氛** 然后直接说：那我们就聊聊技术吧，你是用 **Vue** 栈的对吗，对 **Vue3** 了解的怎么样？

### 自我介绍

**小提示：**自我介绍尽量保持在 2~3 分钟之间；语速均匀，不要结巴；可以多说一下自己非常熟悉的项目，试着将面试官的注意力往自己熟悉的方向拽；

### Vue栈？Vue3熟悉吗

**小提示：**现在貌似用 **Vue** 栈的，大多数情况下都会问问你是否对 **Vue3** 有了解；如果你正好熟悉，可以借此体现技术深度广度；

### 说说Vue3和Vue2的响应式原理吧

**小提示：**经典问题了；回答思路可以按照这个来：

先回答 **Vue2** 的响应式原理，然后点出 **Vue3** 使用 **Proxy** 结合 **Reflect** 代替 **defineProperty**，并且点出有需要的话，可以详细聊了为什么做出这个改变；**诱导面试官深挖；**

这个问题很好体现出 **Vue** 栈的深度（如果Vue能够在一个问题里挖的足够深，就可以不问问了）

[「2022」寒冬下我的万字面试准备【Vue3、Vue2、Vite】篇：【Vue2的响应式】](#)

[「2022」寒冬下我的万字面试准备【Vue3、Vue2、Vite】篇：【Vue3的响应式】](#)

**为什么要用Proxy，区别在哪里；**

**小提示：**先回答 **Vue2** 响应式的缺点；再点明换了 **Proxy** 后的好处；

[「2022」寒冬下我的万字面试准备【Vue3、Vue2、Vite】篇：【defineProperty 和 Proxy 的区别】](#)

**那Vue3为什么要用Reflect呢**

**小提示：**这一题答清 **reflect** 的 **this** 问题修正就可以了，其余的内容看自己的理解进行回答；

[「2022」寒冬下我的万字面试准备【Vue3、Vue2、Vite】篇：【为什么需要 Reflect】](#)

**Vue3/2对数组都做了特殊处理吗**

**小提示：**先回答 **Vue2** 的数组处理，然后再点出 **Vue3** 的数组处理，然后再说明区别；

[「2022」寒冬下我的万字面试准备【Vue3、Vue2、Vite】篇：【Vue3 响应式对数组的处理】](#)

**Vue3相比Vue2除了响应式还有没有别的区别**

**小提示：**可以从几个方面开始聊：

**API特性方面：** Composition API 、 Fragments 多根节点、 SFC CSS 变量、等等都可以

**框架设计方面：** 打包体积更小可以被tree-shaking 、虚拟 DOM 优化、 Diff 算法优化

**兼容性方面：** Vue3 不兼容 IE11 、 Vue3 用 TS 编写，对 TS 支持更好等等

[「2022」寒冬下我的万字面试准备【Vue3、Vue2、Vite】篇：【Vue3 带来的新变化 & 新特性总览】](#)

**来个经典问题：Computed计算属性原理，怎么实现惰性求值**

**小提示：**这一题说清 Computed 怎么实现惰性求值的即可；

[「2022」寒冬下我的万字面试准备【Vue3、Vue2、Vite】篇：【Computed 的实现原理】](#)

**聊聊浏览器原理方面的吧**

面试官问了几个问题，发现 Vue 的理解程度还是比较高的；就不再问了，改为问其余方面的问题，后续面试也没有了 Vue 的问题；

**说说缓存机制**

**小提示：**先聊聊浏览器判断缓存的流程（强缓存优于协商缓存，协商缓存生效就返回 304等等）

然后再说强缓存 Expires 和 Cache-Control 两个缓存标识的区别

然后再说协商缓存 Last-Modified 、 If-Modified-Since 、 ETag 、 If-None-Match 这几个属性

如果你还可以聊，那就讲讲 no-cache 和 no-store 的区别

**最后再点出发启式缓存，留个包袱等面试官问；**

[「2022」寒冬下我的万字面试准备【浏览器原理&安全】篇：【浏览器缓存机制】](#)



## 浏览器解析到JS和CSS时是如何处理的？

**小提示：**先说清：

JS 会阻塞浏览器解析；

CSS 会阻塞浏览器渲染；

CSS 会阻塞后面的 JS 执行；

然后抛出包袱：虽然 JS 会阻塞浏览器解析，但是如果 JS 带上了 `async` 和 `defer` 又不一样；

[「2022」寒冬下我的万字面试准备【浏览器原理&安全】篇：【浏览器解析 不同资源 时的行为】](#)

## JS 带 `async` 和 `defer` 的区别

**小提示：**先说清

`defer` 是延迟执行，`async` 是异步执行

然后再说清 JS 同时带有 `defer` 和 `async` 时 `defer` 不起作用，浏览器行为由 `async` 属性决定。

最后再说清如果同时有多个 JS 带有 `async` 和 `defer`，它们之间的执行顺序关系；

[「2022」寒冬下我的万字面试准备【浏览器原理&安全】篇：【浏览器解析 不同资源 时的行为】](#)

## 如果同时出现多个`async`或`defer`怎么解析；

**小提示：**看上文链接

## 了解哪些浏览器安全方面的问题

**小提示：** `XSS`、`CSRF`、`SQL注入`、`点击劫持`、`DDOS攻击` 这几个都可以聊一聊；

[「2022」寒冬下我的万字面试准备【浏览器原理&安全】篇：【浏览器安全】](#)

## 我们再聊一点计算机网络相关的吧

面试官先是说：浏览器相关的了解的挺多的嘛，那我们再聊一点计算机网络相关的吧；  
转换方向

### HTTP2、HTTP3 聊聊

**小提示：** 二进制分帧 、 头部压缩 、 多路复用 、 服务器推送 ；

这里面抖两个包袱给面试官：我觉得这几个特性里面着重要说一下 多路复用 和 服务器推送 这两个特性，

因为 多路复用 会引出 队头阻塞问题 ， 以及 HTTP3 相关；而 服务器推送 相关的是 103状态码 ；

[「2022」寒冬下我的万字面试准备【计算机网络】篇：【HTTP 2.0】](#)

### 服务器推送这个特性为什么要特别说一下

**小提示：** 先说场景：

2022 年 6 月 Chrome 官方 宣布在 chrome 103 版本对 HTTP 103 状态码提供了支持；

Chrome 官方 也宣布在 chrome 106 版本对 HTTP/2 Server Push 进行禁用；

然后再引诱面试官问为什么要禁用 HTTP/2 Server Push ；

[「2022」寒冬下我的万字面试准备【计算机网络】篇：【详细说说 103 状态码】](#)

### 队头阻塞了解吗？聊一聊；

**小提示：** 回答顺序

先回答 HTTP 队头阻塞，是怎么产生的；

然后回答 HTTP2 怎么解决了 HTTP 队头阻塞；

并且抛出包袱 HTTP2 只解决了 HTTP 队头阻塞，根本的 TCP 队头阻塞还没有解决；

然后就可以聊一聊 HTTP3 ( Quic ) 相关的内容;

[「2022」寒冬下我的万字面试准备【计算机网络】篇：【队头阻塞问题】](#)

## Quic基于UDP如何保证了可靠性?

**小提示:** 先提到 HTTP3.0 , 然后点出它是基于 QUIC 去实现的  
然后最后再说明 QUIC 是如何实现可靠性的;

[「2022」寒冬下我的万字面试准备【计算机网络】篇：【HTTP 3.0】](#)

## 项目问答

上面的基础问的差不多了, 面试官说我们再来聊一点项目吧;

## 监控SDK如何做逻辑解耦 & 多平台?

**小提示:** 这个问题非常的常见, 因为前端监控SDK就得考虑多平台、逻辑解耦的问题;

[说说前端监控平台/监控SDK的架构设计和难点亮点? : 【SDK 如何设计成多平台支持】](#)

## 如何实现插件化 & 按需加载

**小提示:** 插件化就考虑 Core+Plugin 的设计, 按需加载就考虑将插件单独打成 npm 包

[说说前端监控平台/监控SDK的架构设计和难点亮点? : 【SDK 如何方便的进行业务拓展和定制】](#)

大体回答:

- 先将 SDK 内的代码抽离成 Core+Plugin 的形式
- 然后按需加载考虑将每个 plugin 单独打包, 让用户自己引入;

## 监控项目的上报策略和上报优化

**小提示：**这个问题可以从 [上报方式](#)、[上报时机](#)、[其余上报优化](#) 分点回答；

说说前端监控平台/监控SDK的架构设计和难点亮点？：[【上报策略】](#)

## requestIdleCallback 和 requestAnimationFrame区别

面试官听到我提到了 `requestIdleCallback`，多问了一个问题。（笑哭）

### requestIdleCallback

- `requestIdleCallback` 是一个兼容性不那么好的功能，所以我们使用前得判断它是否支持
- 我们可以使用 `window.requestIdleCallback()` 方法来插入一个函数，这个函数将在浏览器空闲时被调用；`requestIdleCallback`
- 它的参数为 `callback` 和 可选的 `timeout`；如果指定了 `timeout` 且为正值；则回调在 `timeout` 毫秒后还没调用时，回调任务就会被放入事件循环里排队，这样做可能会影响性能；

注意点：因为它发生在一帧的最后，此时页面布局已经完成，所以不建议在 `requestIdleCallback` 里再操作 `DOM`，这样会导致页面再次重绘。

### requestAnimationFrame

- `window.requestAnimationFrame()` 告诉浏览器——你希望执行一个动画，并且要求浏览器在下次重绘之前调用指定的回调函数更新动画。
- 该方法需要传入一个回调函数作为参数，该回调函数会在浏览器下一次重绘之前执行

## 为什么不用 gif 图片上报而是用 beacon

**小提示：**可以从低优先级的请求、浏览器关闭不会导致上报失败、不占用主要任务等等方面解释

## 笔试

[509. 斐波那契数](#)；这里要求用数组存值减少计算；

## [46. 全排列](#)

## [11. 盛最多水的容器](#)

## 二面技术面

### 活跃气氛的聊天

先随便唠了唠活跃了下气氛，然后让做个简单的自我介绍；之后就开始问答；

### 削峰限流的实现逻辑

[说说前端监控平台/监控SDK的架构设计和难点亮点？](#)：【[削峰限流](#)】

### 监控告警是怎么实现的

[说说前端监控平台/监控SDK的架构设计和难点亮点？](#)：【[监控告警](#)】

### 服务端错误聚合的实现方式

**小提示：**前端通过生成错误hash值，判断是否同一个错误

[说说前端监控平台/监控SDK的架构设计和难点亮点？](#)：【[错误HASH生成](#)】

### 是否有考虑到错误信息相同，但是不是同一个错误的场景

**小提示：**这一题同上面的链接，采用错误堆栈聚合解析的方式；

### message是动态变化的，但是实际上是属于同一个错误怎么处理？

这个问题问的很细，面试官的意思是：

假设说我们的 `message` 是一个动态变化的，有的场景比如 `a.b.c`；这里的 `b`、`c` 属性是动态变化的，那么如果 `a.b` 为 `undefined`，`c` 就会报错；但是根据 `c` 的值不同，

message 也会不一样;

这个问题我是没有太好的方案，不过面试官给出的 解决逻辑 是：

前端获取到错误信息时，不生成错误 hash，而是原样发送给后端，由后端采用一个 字符串相似度 算法，判断能不能 聚合 到一起；

## 聊聊性能监控相关的吧

面试官可能是感觉这个问题差不多了，就转换了话题；

## 采集了哪些性能指标

**小提示：**我回答这个问题，都会先回答：我将性能指标分为 以用户为中心的性能指标 和 以技术为中心的性能指标，然后再分开聊；

[一文摸清前端监控自研实践（一）性能监控：【以用户为中心的性能指标】](#)

## FCP 和 FP的区别是什么

[一文摸清前端监控自研实践（一）性能监控：【白屏（FP）、灰屏（FCP）】](#)

## 如何优化FCP的值

见上面的性能优化题目

- 重定向优化：
  - 避免重定向
- DNS、TCP 优化：
  - 预解析；也就是用 `<link rel="preconnect" href="//`sample.com`" crossorigin>`
  - 使用 `preconnect` 时，同时也使用 `dns-prefetch` 作为兼容手段，或者凸显关键资源；
- 请求优化：
  - 减少 HTTP 请求数量：多个压缩成一个；
  - 使用 HTTP2：头部压缩

- 减少 cookie 大小: http 请求会携带 cookie
- DOM、CSS 解析优化:
  - CSS 放在头部
  - 优化 CSS 的 选择器 结构: 浏览器解析 CSS 选择器是从右到左;
  - JS 放在尾部, 或者标注 defer、async 避免阻塞解析;
  - 精简 DOM 结构
  - 提取 critical css: 也就是将首屏需要用到的 CSS 放在 Head 里加载, 其余的异步加载;

## 有没有采集 FMP, 如果要你采集会怎么做;

FMP 指标 (主要内容可见时间), 是 google Lighthouse 提出的概念;

- 它有多重算法, 我们可以认为 FMP 的时间点是页面渲染过程中元素增量最大的点, 然后对每个元素, 根据其是否可见、对页面的影响程度等设置不同的权重, 然后计算权重得分;
- 同样, 我们也可以通过手动埋点的方式获取首屏时间;
- 但是其实首屏统计本身就是浏览器的职责, 把它交给浏览器去处理是最好的, 目前 W3C 关于首屏统计已经进入了提议阶段, 可以等待 W3C 再次标准化,
- 而现在可以直接获取到的指标中有 LCP, 它也是页面加载过程中增量最大的时间点, 一定程度上就可以以它作为首屏时间; (MutationObserver API 可以用在这里)

## long task 指标怎么采集?

**小提示:** longtask 的监听比较麻烦

因为用 PerformanceObserver 只能采集到 longtask 的开始时间和耗时, 是哪个任务产生的并无法知道;

所以我们需要采集用户的所有行为数据 (某个时间点点了什么), 画成时间轴, 结合 longtask 的区间来进行分析;

## 聊聊在线协作项目相关的把

这个项目更多的是架构设计方面的东西, 因为跟公司业务强相关, 问题答案就不细写了;

问题包括:

- 聊聊在线协作项目的技术选型
- 在线协作项目的架构设计
- 每个分层之间的细节设计实现
- 断线重连的实现考虑，正确性保障；
- 如何保证 OP 消费有序且正确
- 渲染方面的优化 & Canvas 优化
- OT 冲突处理架构

## 笔试

### [121. 买卖股票的最佳时机](#)

### [122. 买卖股票的最佳时机 II](#)

深拷贝，要求处理 基本数据类型、引用数据类型（要继承原型上的数据）、Set、Map、解决循环引用；

## 三面技术面

三面录音没录上，所以这里就没法给出问题了；但是 大多数情况下是在聊天；

- 聊聊场景实现
- 聊聊项目的架构
- 项目的未来发展
- 个人的职业看法等等

## HR面

HR面的问题比这个要多，但是跟三面一样，录音没有录上，我简单列几个问题；

- 为什么要离职
- 你期望什么样的团队和工作内容？
- 你平时是怎么学习新技术的？
- 当前薪资
- 期望工作地点
- .....



## 一面电话面

电话聊+笔试+最后反问 总计 1.5h

## 自我介绍

**小提示：**说实话，自我介绍说多了，都可以流利的直接背下来了，一大段话在两分钟内；

## 前端参数配置有没有做成动态取的（从后端）

**小提示：**面试官的意思就是，如果配置成HTTP方式配置，就可以变成动态配置方式；

直接回答了没有，我给的原因是如果 **SDK** 初始化时通过 **HTTP** 去拉配置表，可能会造成主应用的性能损耗；

## 数据清洗的实现逻辑

[说说前端监控平台/监控SDK的架构设计和难点亮点？](#)：【数据清洗】

## 削峰限流的实现逻辑

[说说前端监控平台/监控SDK的架构设计和难点亮点？](#)：【削峰限流】

## 前端暴力削峰有没有考虑以用户为比例；

**小提示：**这个问题确实我有一点意外；原先都没有考虑过前端的削峰限流直接以用户为维度进行处理；

**人话：**在初始化 **SDK** 的时候，就 **Math.random** 来决定这个用户是否需要上报；这样避免了有一些用户的数据上报一部分后，数据链路追踪不完整的问题；

原先确实没有考虑到，聊后有所收获；

## 监控哪些性能指标

**小提示：**我回答这个问题，都会先回答：我将性能指标分为 **以用户为中心的性能指标** 和 **以技术为中心的性能指标**，然后再分开聊；

### [一文摸清前端监控自研实践（一）性能监控：【以用户为中心的性能指标】](#)

## 每个性能指标怎么采集

**小提示：**看上文，不贴链接了；

## LCP的值，如果被骨架屏干扰怎么办；

这个问题一开始确实懵了几秒钟，之前都没有考虑过这个问题；然后就跟面试官讨论可行的方案，讨论了好几分钟；

**我提出的方案：**类似于用 **mutationobserver** 采集 **首屏时间** 的逻辑，采集 **DOM** 元素变化最大的点，并且让骨架屏的 **DOM** 在权重里面为 **0**；就可以忽略骨架屏对于 **LCP** 的影响；

最后面试完后研究了一下，发现官方对于 **LCP** 的取值，本身就不取骨架屏的 **DOM** 元素，它只取 **文本**、**img** 等有效元素所在的 **DOM**（笑哭）；

## 上报PV还附带哪些数据

**小提示：**聊到上报 **PV** 时，会附带将各种元数据带上去，包括 **当前页面信息**、**用户来路**、**用户跳转方式**、**用户UID**、**会话ID**、**登录ID**；

## 如何重写底层的 **XMLHttpRequest**

### [一文摸清前端监控自研实践（二）行为监控：【XMLHttpRequest 的劫持】](#)

## 在线协作项目相关

这个项目更多的是 架构设计 方面的东西，因为跟公司业务强相关，问题答案就不细写了；

### 包括了下面这些问题：

- 在线协作表格的整体架构设计
- 每个分层之间的细节设计实现、职责

### 笔试：

上面的聊完后，面试官说发个笔试链接给我，我们做几个题，然后就把链接发到我的邮箱，打开三道笔试题；

- 监听页面的所有 `a` 标签 `click` 事件，判断 `href` 是否以 `https` 开头；
- 手写 `Event` 事件，包括 `on`、`off`、`once`、`trigger`
- 爬楼梯，LeetCode原题，但是我忘了是 [70. 爬楼梯](#) 还是 [746. 使用最小花费爬楼梯](#)

## 二面技术面

视频聊+最后反问 总计 1h

### 自我介绍

**小提示：**自我介绍尽量保持在 2~3 分钟之间；语速均匀，不要结巴；可以多说一下自己非常熟悉的项目，试着将面试官的注意力往自己熟悉的方向拽；

### 为什么要自研监控平台

**小提示：**因为目前市面上成熟的监控产品很多，如果你是自研的监控平台，势必会被问道这个问题，可以提前先总结归纳一下；

我的大概回答：

- 可以实现前后端全链路API监控
- 接入公司缺陷管理平台（bug 平台，实现一系列的监控告警闭环流程）
- 可以自定义业务的监控维度和数据分析维度（一些具有独特监控角度的业务，需要自研）

这里面试官比较在意的点在于：自研相较于使用 Sentry 是否更有意义价值，自研的契机是什么，讨论了很久；

## 全链路API监控是做了什么

## SDK在设计时考虑了哪些问题

小提示：可以先看我之前文章中总结的点

[说说前端监控平台/监控SDK的架构设计和难点亮点？](#)：【值得一谈的点】

我的大概回答：

- 如何设计成支持多平台、可拓展、可插拔的 SDK
- 如何设计 SDK 的上报策略

## SDK的上报策略、上报时机

小提示：可以先看我之前文章中总结的点

[说说前端监控平台/监控SDK的架构设计和难点亮点？](#)：【上报策略】

## 为什么不用gif图片，跨域怎么办

小提示：上面回答了上报策略后，面试官追问为什么不用 gif 图片、跨域 怎么处理；

- Image 是以 GET 方式请求图片资源的方式，将上报数据附在 URL 上携带到服务端，而 URL 地址的长度是有一定限制的。规范对 URL 长度并没有要求，但是浏览器、服务器、代

理服务器都对 URL 长度有要求。有的浏览器要求 URL 中 path 部分不超过 2048，这就导致有些请求会发送不完全。

- 至于 跨域 问题，作为接受数据上报的服务端，允许跨域是理所应当的；

## 有没有考虑用gif分片上报

**小提示：**因为前面提到 xmlhttprequest 进行合并上报，作为 sendbeacon 的兜底方案；就问有没有考虑不是使用 xmlhttprequest 而是采用分片的方式进行上报；

## 上报失败怎么处理

**小提示：**这种题目，就将自己的实现、思考都说出来即可；

### 我的大概回答：

- 如果请求失败，会将数据隔几秒重传一次，如果再次失败；就中止所有的上报；将数据存入 localStorage ；
- 下次进入时，判断 localStorage 要不要进行重传；

## 采集了哪些性能指标

**小提示：**我回答这个问题，都会先回答：我将性能指标分为 以用户为中心的性能指标 和 以技术为中心的性能指标，然后再分开聊；

### [一文摸清前端监控自研实践（一）性能监控：【以用户为中心的性能指标】](#)

## 性能指标评分是怎么做的

**小提示：**这种问题，先说明不能对所有的应用设置统一的基准线  
然后再说说不同指标的权重不同，分数占比不同等；

### 我的大致回答：

- 首先有一个结论，那就是对于所有的应用设置统一的性能指标基准线是没有意义的；因为不同项目的体量、设计都是不同的，标准也就不同；
- 所以说我们在平台上有一个设计，那就是可以在界面上设置具体指标的评分基准线；我们会有一个默认的性能指标基准线以及指标比重；指标包括：FP、LCP、CLS、FID 这几个以用户为中心的性能指标，然后用户可以自己在默认基准线和比重的基础上进行调整；
- 然后最后会根据设置的基准线进行计算得出评分；

## 错误聚合怎么实现

**小提示：**前端通过生成错误 hash 值，判断是否同一个错误

[说说前端监控平台/监控SDK的架构设计和难点亮点？](#)：【错误HASH生成】

## 告警逻辑 & 策略

**小提示：**提到了告警逻辑有宏观告警和微观告警，还追问微观告警是否会带来很多误报情况（网络情况 等）

[说说前端监控平台/监控SDK的架构设计和难点亮点？](#)：【如何设计监控告警的维度】

## 在线协作项目细节 & 架构设计 & 难度深挖

这个项目更多的是架构设计方面的东西，因为跟公司业务强相关，问题答案就不细写了；

问题包括：

- 整体项目的架构设计分层
- 聊聊技术选型
- 自己的角色 & 负责的架构分层
- 网络层的职责 & 设计
- 网络层的难点 & 挑战 & 解决
- 如何保证 OP 消费有序且正确
- OT 实现的 算法逻辑 & 架构方案逻辑
- 为什么要做在线协作项目，能给公司带来什么产出；

## 有什么想要问我的

**小提示：**在面试的最后，抓住这个机会问一些内容，包括说团队、技术、等等；

### 我的大致回答：

- 介绍一下团队和团队业务
- 想要招一个什么样的人；
- 对我的面试印象是怎么样

## HR面

HR面我就列举一下具体的问题，没有答案了哈，总计 1h

## 聊天

## 简单自我介绍

## 毕业时对读研和直接工作的考虑

**追问：**跟父母沟通读研和工作时是否有意见不统一的时候，怎么解决；

## 本科的时候有没有什么实习经历

**回答：**没有实习，但是有在学校研究实验室有过经历

**追问：**导师是怎么评价你的

**追问：**导师让你做前端你是什么想法

**追问：**导师让你做研究的时候有没有不开心、困难、挫折的事情

**追问：**遇到了挫折的事情后有没有做什么总结反省呢

## 你现在所在的公司的收获是什么

为什么想要离开呢

心中对阿里、对岗位的期待

业余爱好

使用淘宝的时候，有什么可以觉得做得更好

回答：我举例了淘宝购物车的例子，购物车扩容；

追问：还有什么觉得淘宝没改，或者改没有改好的地方吗

追问：觉得拼多多、京东有什么做得比淘宝更好的地方吗

觉得这一段时间，对于阿里巴巴来说过的不顺，比如股价跌、垄断，你怎么看待这个事情

你觉得自己未来还有什么可以提升的，在非技术层面

换工作、换城市会跟父母聊吗，他们怎么看待这件事

现在的薪资是多少

你觉得什么样的技术团队是你期待的

你还有什么想问我的吗