

面试

半小时内，面试官爱上你

为什么项目没有亮点，任何一个需求，只要把数据量想的贼大，就是亮点

1. 反对没事就刷题
2. 核心知识体系

面试不好拿高薪，其实一各面试题，就可以分出级别

1. 文件上传
 1. 大部分同学都做过 没有亮点
2. 输入url到vue渲染页面发生了啥
3. 表格渲染

文件上传这个题，就可以把前端大概的分成

代码地址 <https://github.com/shengxinjing/file-upload>

1. 菜鸟

1. axios.post + 上传进度条 + 后端node接手文件
2. 相亲的妹子问你，你有什么擅长的，： 小学三好学生

2. 10K

1. 拖拽 粘贴

3. 15K

1. 分片 断点续传 (开始得分)

4. 20K

1. 如何限制只能上传png格式的

1. .split() 基本不得分的 因为.pdf文件重命名 就超过了

2. 文件头信息

2. 如何判断文件的唯一性

1. 算hash 比如md5

2. 文件过大，1各G，怎么算md5 不卡顿

1. web-worker

2. 时间切片（学了react fiber架构get到的知识点，requestIdleCallback）

3. 抽样

5. 25K

1. 文件切片数量过多，上传请求过多，导致卡顿，怎么处理

1. 控制异步任务并发数量（本来一个头条面试题）

6. 各种场景的深挖

1. 如果又报错，怎么重试，报错三次怎么统一终止所有切片

7. 每个的切片的大小如何控制，移动端和pc端网速不一样的

1. 应该根据网速来动态调整包的大小

2. TCP协议的慢启动

8. 并发数+包控制如何实现？

、、、

1. 全栈才是未来

2. 原理才是硬道理

1. 值钱的技能，都是不好学习的

关键有的公司不问值钱的问题

这家公司别去

小公司的面试，五花八门 需求多样化

大公司的面试，基本都一样 体系化的知识，而不是具体某个需求

学历很重要，如果不能提升，就搞开源，秀秀肌肉

面试造火箭，工作拧螺丝是真的吗

确实是真的

1. 竞争比较多
2. 早火箭，就是一个合格的工程师应该具备的
3. 就算现在拧螺丝，也要又造火箭的心

应届生， 刷算法 leetcode

好的面试: 只问你会的， 问到你不会

技术热点: 不要盲目追热点， 技术的本质一直都是不便的

算法， 数据结构， 设计模式， 计算机组成

1. 虚拟dom

1. 树这个数据结构

2. fiber

1. 时间切片， 借鉴了cpu调度的逻辑

2. 链表这各数据结

3. vue模板编译

1. 编译原理的js实现

2. v-if => 编译成三元表达式

从用户输入url 到vue项目显示，发生了啥

1. 域名=? ip地址, dns协议
2. 建立TCP链接 发请求
3. 后端接受请求, 读文件, 读数据库, 获取网络接口, 最终返回想要的数据和页面
4. 前端接受html 渲染页面
 1. html=>dom
 2. css=> css tree
 3. dom+css tree => rendering tree
 4. 渲染页面
5. 执行js

1. 网络协议
 1. TCP IP HTTP HTTPS DNS啥关系
 2. http的缓存
2. 浏览器的渲染机制
 1. how browser works
3. 框架执行的侧面

1. Vue or React
2. 同构ssr的情况
4. V8

1. 问多的不多

性能优化

技术特点：

一定要写亮亮点 写自己会的

纯技术点面试，闭包，原型链，红宝书 确实没啥扩展的
开放式才是好的

1. 怎么设计一个弹窗组件

1. 怎么做文件上传
2. v-if到底咋执行的

啥事亮点

1. 我会htmlcss 这个不是
2. 我会vue 这个不是
3. 我会node 这个不是
4. 任何皮毛的 董不是， 你觉得你旁边的哥们也会， 那

就不是亮点的

1. 原理源码 会vue源码，用element不算，element源码算
 1. react使用不算，懂vdom 懂hooks，懂setState才算
2. 解决过高难度的问题
 1. 流量大，数据量大
3. 全栈
- 4.

1. 熟悉组件化设计，自己实现过element相关组件
2. 熟悉vuejs 看过源码
3. 熟悉react 看过源码

I wrote python

如果没有亮点怎么办

好好学习

熟悉源码，并不是只看过一些文章，真正的看过

以是否能造出轮子，来衡量学习结果

任何简单的技能，都不值钱 我们就要学男的东西，才值钱

VUe和react虚拟dom的区别

只渲染课件的3屏或者5屏，模拟滚动条就可以了