

三面

考察前端架构能力、前端工程化的理解，从解决难的问题和解决快的问题两方面考察

框架实现

- vue/react 的实现原理，可以结合实际场景看这些原理的使用场景
- 实现 v-for 指令，有点复杂，大概的实现方式类似将 v-for 中的内容作为一个子组件，接收的 props 为 list 当前项的值，然后生产这个子组件
 - 什么是事件委托（代理），其原理是什么，好处体现在哪些方面，在 v-for 中有什么应用场景
 - 通过事件委托，将 v-for 中的所有事件通过父组件代理
- Vue Router 导航解析的具体流程是什么样的，使用过哪些导航钩子，分别在什么场景下使用
- Vue 的响应式原理中 Object.defineProperty 有什么缺陷
- v-modal 的实现原理，如何在自定义组件中使用
- 什么是 virtual dom，其真的比原生 dom 快吗
- diff 算法并不稀奇，而为什么在 react 使用中 diff 算法对性能优化如此重要（传统 diff 算法与 react 的 diff 算法有何区别）
- React 写的好好的，为什么突然要搞一个 Hooks？（考察对 react 发展，和对组件拆分方便性的理解）
- React Router 4 为什么引入基于 component 的动态路由概念，而不再用之前配置路由的方式？
- React Router 4 中将组件按需加载如何实现？React Router 5 相比于 4 新加了哪些功能？
- useMemo、useCallback 使用场景，它们的实现原理。

大型项目架构能力

- 开始一个项目的时候需要首先考虑哪些东西，如何处理这些
 - 登录注册，模拟登录注册
 - 接口封装，接口约定，mock 数据
 - layout 布局
 - 前置信息准备，放在哪里做比较合适
 - 页面级组件
 - 公共组件 -> 如何为开源组件做准备
 - 基础样式规范 -> 感情色、主题色、间距、字体
 - 路由渲染
 - 权限控制及实现
- 架构及业务经验
 - 常用 loader，项目中自己引入的 loader，打包时做了哪些优化。
 - 新版 babel 更新后的各配置含义
 - 打包 es6 转义 es5 时，class 类会用什么方式转化
 - 如何设计一个用户权限系统（或说一说前公司的权限系统业务逻辑，群组，角色等概念）可以进一步扩展到如何进行密文校验，存储，密文加密策略，密文类型，对称加密非对称加密的方式等。
 - 实现动态更新主题
 - ie 浏览器适配遇到过哪些坑

前端工程化

主要目的为了提高效率和降低成本，从模块化、组件化、规范化、自动化4个方面去思考。通过这四个方面达到软件生产的有序，流水线作业。

1. 模块化

- 模块化就是把一个大的文件，拆分成多个相互依赖的小文件，按一个个模块来划分
- 让前端可以构建大型应用，实现代码复用，提高代码可读性
- amd/cmd

2. 组件化

- 页面上所有的东西都可以看成组件，页面是个大型组件，可以拆成若干个中型组件，然后中型组件还可以再拆，拆成若干小型组件组件化≠模块化。
- 模块化只是在文件层面上，对代码和资源的拆分；
- 组件化是在设计层面上，对于UI的拆分目前市场上的组件化的框架，主要的有Vue, React, Angular2

3. 规范化

- 在项目规划初期制定的好坏对于后期的开发有一定影响。包括的规范有
- 目录结构的制定
- 编码规范
- 前后端接口规范
- 文档规范
- 组件管理
- Git分支管理
- Commit描述规范
- 定期codeReview
- 视觉图标规范

4. 自动化

- 也就是简单重复的工作交给机器来做，自动化也就是有很多自动化工具代替我们来完成
- 例如持续集成、自动化构建、自动化部署、自动化测试等等

针对上述答案问答，看候选人对工程化整体到细节的理解

- 什么是前端工程化
- 如何建立起这一套体系
- 针对上述任一模块或关键点展开讨论

CICD、运维部署、docker、网络

- nginx 使用，如何配置如下信息，具体到字段名，字段值
 - proxy_pass, auto redirect, cors, gzip, cache-control, location
- deploy, ssh
- http&https区别、sql注入、xss、csrf攻击

前端趋势、新技术

- 了解过哪些新的前端技术，有实践过哪些
- resetful、graghQl区别 应用场景
- webassembly做什么的，能有哪些场景用到
- 微服务、serverless都是做什么的，各自适合什么场景
- 微前端概念，如何实现
- Jamstack 是啥

英语阅读

- 考察是否经常阅读英文文档，方便定位问题

其他

- 数据埋点有哪些方式，如何实现
 - 代码埋点、可视化埋点（如何实现）
- 在工作中发现哪些问题，提升方式有哪些方面
 - 开放式的问题，主要考察有没有发现问题，解决问题的思维模式
- 情绪考察，同事间如前后端配合不愉快时如何面对，遇到情绪问题时的处理方式，项目验收，工作压力比较大时如何对待。