

人资项目面试逐字稿

01

问: 实际开发中有用过环境变量吗?

02

问: axios有没有做过一些业务封装?

问: 工作里中的token是怎么管理的?

03

问: 业务中的跨域问题是如何解决的?

04

问: 怎么根据token的有无去控制路由的跳转?

问: token有一定的失效性, 过期了该怎么做?

06

问: 工作中有做过数据处理吗?

08

问: excel导入功能的实现流程简要描述一下?

问: 有没有封装过一些自己的业务组件, 简单说一下?

问: 有没有开发过一些vue插件? 举例说说

问: excel导出功能的实现流程说一下?

09

问: 有封装过v-model相关的双向绑定组件吗?

10

问: 有做过权限相关的事情吗?

回答问题的重点

1. 明确面试官的问题 不清楚可以追问 避免回答不在点上
2. 回答问题有条理有逻辑 建议按照总分方式
3. 技术问题最好有理有据 言之有物 先说场景 再说思路 再说具体解决方案

注意: 以下面试题按照人资项目的模块维度进行划分, 如果看逐字稿不太清楚当时是怎么讲的, 可以打开对应的模块进行快速复习!

01

问: 实际开发中有用过环境变量吗?

有用过的, 我在公司里用过的场景主要是为了和后台对接接口的时候切换根域名baseURL时用, 因为公司有几个不同的环境嘛, 一个环境对应前端一个环境变量文件, 同样一个环境变量比如说VUE_APP_URL可以在不同的文件里配置不同的接口地址, 再配合运行命令配置启动时指定一下运行的环境变量文件很容易就切过去了

02

问: axios有没有做过一些业务封装?

啊, 有的, 在项目里面我主要是封装过axios的拦截器部分

我在请求拦截器里面做了注入全局token的事情, 这个事儿是因为有很多接口都需要这个token来做数据鉴权, 为了避免书写多次 统一配置了一下

响应拦截器的部分, 我主要是做了一个自定义的成功错误判断 这个事儿, 可以简单说一下, 因为我们知道 拦截器的成功和失败判断本身是通过 http 状态码 200- 300 但是我们后端业务状态比较多一些 他们并没有采取这种依据作为判断条件, 这样会有问题, 导致前后端逻辑对不上 最重要的问题是 发生接口错误 感知不到 我做的事儿就是在成功回调里 自己做了判断 成功就返回数据 失败就手动return Promise.reject()

问: 工作里中的token是怎么管理的?

我们公司的token管理都是通过vuex配合本地存储来做的, 使用vuex是因为token数据比较特殊, 在很多模块中都可能会用到, vuex方便管理, 配合本地存储比如localStorage, 是因为vuex有一个不好的地方就是刷新就丢失, 因为它是基于内存嘛, 配合一下localStorage可以保持持久化

追问: 如果不用vuex只用本地存储能实现吗?

答: 效果上自然是没有问题的, 不过我感觉俩种内存的优势融合一下岂不是更好吗

03

问: 业务中的跨域问题是如何解决的?

这个的话我们公司大概分了俩种环境，一种是开发的时候，一种是上生产的时候，开发的时候因为要对接的后端可能会比较多，他们配置不太方便，这个时候采取的是比较方便的前端代理的方案，和谁对接的时候就把target设置为对方的接口ip 解决跨域问题 要是上生产的话 通常就是后端来配置一个服务 cors

追问: proxy代理方案的原理是什么?

跨域的产生原因其实就是三要素不一致，加上浏览器同源策略导致的，如果我可以在前端和接口端之间架设一台中间服务器，保证它和前端三要素一致，和接口端因为是服务对服务不走浏览器 所以不跨域 就可以解决这个问题

追问: 那你了解跨域产生时，数据到底有没有发出去?

事实上数据是能发送到服务端的，那边也能接收到请求，只不过数据在返回客户端的时候被浏览器给拦截了

04

问: 怎么根据token的有无去控制路由的跳转?

vue这边的路由自带了路由前置守卫，我们可以在前置守卫里拿到token数据，然后根据需求做分支判断，要是token存在就使用next方法正常放行跳转，否则可以强制跳回到登录，让用户去获取token

问: token有一定的失效性，过期了该怎么做?

我这边采取的是被动失效，主要是后端失效之后会返回前端一个401的状态码，我们可以在axios的响应拦截器里，去获取这个状态，如果发现是401就清理一下过去的token，然后重新跳转到登录页，获取有效token

06

问: 工作中有做过数据处理吗?

主要是在组件和后端返回的数据之间, 或者组件产生的数据和需要提交给后端的数据之间, 有可能会出现结构对不上, 这个时候可能会处理一下, 举个例子, 比如说我们常用的tree型组件要求必须是嵌套的tree型数组, 这个时候就需要处理一下, 再比如提交的时候接口要求的是字符串, 我们组件给到的数组, 也需要处理

追问: 既然说到了树形处理, 说一下处理的思路

树形数据处理, 通常后端那边返回的格式是一个普通的平铺结构, 不过一般都会有一个pid字段, 可以根据它找到自己的父节点, 那办法就有了, 或者使用递归的方式, 性能好一点的话可以采用非递归的方式都可以, 主要还是一个寻找父节点的过程

继续追问: 详细说说呢, 伪代码思路是怎么样子的?

我们可以按照下面的步骤去写这个处理函数

1. 可以先遍历原数组以它的id作为key, 本身作为value形成一个对象结构
2. 再次遍历原数组, 通过每一项的pid字段取第一步形成的对象中尝试进行对象取值, 如果取到了值就把自己push到它children属性中, 如果找不到代表它本身就是最外层的父节点, 直接push到函数最终产出的数组中

08

问: excel导入功能的实现流程简要描述一下?

当时公司的场景大概一个excel文件里就几十条数据, 量比较少, 和后端商量之后制定了前端主导的方案, 解析的过程放到了浏览器端做, 当时是参考了一下vue-admin中的现成的方案

大概流程是这样的, 人事那边准备好一个标准的excel文件, 然后通过插件解析成js数据, 把js数据格式化一下, 最后通过调用后端接口完成导入

追问: 如果数据量比较大, 你打算怎么做?

emm... 如果量很大的话，浏览器算力扛不住，可能在技术选型的过程中就让后端主导了，前端只负责一个文件上传，把excel文件传过去之后让服务端做解析的事情，前端还省事了

问: 有没有封装过一些自己的业务组件，简单说一下？

有的，不过我封装的基本都是和业务挂钩的组件，我挑一个比较简单的说一下，之前做业务的时候有遇到过 多个业务模块用到了类似的模板结构，就使用插槽封装了一些组件

大概的封装思路就是先把可复用的模板部分写好，然后把可变的需要自定义的部分用slot插槽占位 要是插槽个数比较多的话，可以区分一下默认插槽和具名插槽，尤其是功能区域比较明显的部分，用具名会更加清晰

问: 有没有开发过一些vue插件？举例说说

有过，项目开发的时间长了，沉淀了不少业务通用全局组件，想把他们统一进行注册，就封装了一个小插件

当时其实一开始也没有什么思路，后来扒了一下 elementUI的源码，仿了一下它的写法，流程我还大概记得

先是把所有的组件放到一个数组中，然后在插件的install方法里遍历整个数组调用Vue.component方法依次注册

最后在main.js里引入插件，通过Vue.use()进行插件注册

追问: 你如何理解vue插件，什么事儿可以用插件来做？

其实我理解的所谓插件就是vue做一些全局的拓展，增加一些它本来没有的功能

你比如添加一些静态的全局方法，再比如在原型上挂载一些实例属性，或者注册一些全局组件或者指令都算

总的说来，反正只有你能通过Vue构造函数能做到的事儿基本上都可以搞成插件

问: excel导出功能的实现流程说一下?

导出的话, 我们因为到处的数据量不大, 所以直接采取的时候前端主导的方案, 参考的现成方案实现的导出

大概得流程就是

1. 调用后端接口得到要导出的数据
2. 把数据简单处理一下转化成导出插件需要的格式
3. 把数据配置到对应的位置就行

插件会自动帮助我们生成excel文件并且自动下载到本地

追问: 那如果要导出的数据量比较大怎么解决?

那可能一开始就要采用后端主导的方案了, 我们可以把这个excel文件生成的过程放到服务器去做, 前端只需要调用一个下载接口, 把后端准备好的excel文件下载到本地就行

09

问: 有封装过v-model相关的双向绑定组件吗?

有的, 之前在做一个通用的上传组件的时候, 一开始是通过传统的父子通信实现的, 父传子实现的图片回显, 子传父实现的图片url抛出传给接口, 后来使用v-model做了一点优化, 简单了不少

追问: 具体是怎么做的, 大概说一下步骤

我想想哈, 大概是我们知道v-model是一个语法糖, 当你在一个组件上写了v-model绑定了某个字段, 这个时候其实它悄悄做了三个事儿

1. 给组件绑定了value的属性
2. 给组件绑定了input的事件
3. 事件触发时参数的自动赋值

那基于这三个事儿, 我们就可以把原本需要自己绑定给组件的prop干掉, 直接去组件里接受value的属性, 也可以把之前需要手动绑定的事件干掉, 直接在组件中通过\$emit触发input事件, 甚至最后事件传出来的最新的值赋值的事儿也不用干了

问: 有做过权限相关的事情吗?

有做过, 我们前端这边主要做的就是菜单权限控制和按钮显示控制

追问: 说说菜单权限怎么做的?

权限设计是一个需要前后端配合的事情, 大概分成下面几个步骤

1. 项目通过几个业务模块的配合生成对应的权限数据
2. 把后端返回的权限数据和前端本地的路由做对比, 得到过滤之后的有资格显示的路由数组
3. 调用路由核心方法`router.addRoutes`把路由数组加入到路由系统中
4. 如果需要显示到左侧菜单里, 可以配合`vuex`做, `vuex`存一份相同的数据, 渲染左侧的菜单

追问: 按钮的权限又怎么做呢?

按钮的权限其实就是控制显示隐藏

思路就是通过后端返回的权限标识和按钮自身的标识`code`做对比, 如果有资格就显示, 否则就隐藏

这里根据思路我们可以封装一个全局指令, 通过指令可以实现复用, 就可以在需要控制的按钮身上进行指令绑定, 从而控制按钮的显示也隐藏