

记录一次笔试题 (2021.0514)

- 一年工作经验，杭州面试。

人事面

- 能接受的加班情况
- 为什么想换工作
- 期望工资范围
- 未来有什么规划

技术面

- 自我介绍，在学校时的情况，最近做的什么项目，使用技术栈？
-

正题，Vue 相关(按分类归纳，非面试询问顺序)

一、平时工作Vue.\$set()是怎么用的？

- 当把一个普通的 JS 对象传入 Vue 实例作为 data 选项，Vue 将遍历此对象所有的属性，并使用 Object.defineProperty 把这些属性全部转为 getter/setter。
- 官方定义：如果在实例创建之后添加新的属性到实例上，它不会触发视图更新。
- Object.defineProperty 的缺点：无法检测对象属性的添加和删除
- 可以使用Vue.set(object, key, value)方法将响应属性添加到嵌套的对象上

二、Vue中data为什么必须是一个函数呢？

- 首先data中存放的很多数据，指向同一个地址
- 页面中每个组件的值都是从data中获取的（同一个地址）
- 当data使用一个函数返回时，每个实例的data属性都是独立的，不会相互影响。
- 总结就是避免多个组件共用同一个作用域data。

三、在Vue生命周期运用、理解及操作？

- beforeCreate时，只是始化一个空的Vue实例对象，并未初始化data数据和methods方法。
- created时，data和methods已经初始化完毕，可以在这里进行数据操作或调用methods方法
- beforeMount时，模板在内容中编译好了，但未挂载到页面上（此时页面插值展示{{msg}}）
- mounted时，Vue实例已经初始化完毕，浏览器页面正常展示。
- data数据发生改变时
 - beforeUpdate：此时data中数据已经更新，但是页面上还是旧数据。
 - updated：此时页面上数据已经与data数据保持一致，都是最新的。
- beforeUpdate时，未进入真正销毁过程，实例身上挂载的data和methods都可以继续使用
- destroyed时，组件已被完全销毁，组件数据及方法指令都不可在用。

四、当Vue中组件嵌套时，它们生命周期关系是怎样的？

- 组件正常加载渲染过程 父beforeCreate->父created->父beforeMount->子beforeCreate->子created->子beforeMount->子mounted->父mounted

- 子组件更新过程 父beforeUpdate->子beforeUpdate->子updated->父updated
- 父组件更新过程 父beforeUpdate->父updated
- 销毁过程 父beforeDestroy->子beforeDestroy->子destroyed->父destroyed
- 可知子组件挂载完成后，父组件还未挂载。所以组件数据回显的时候，在父组件mounted中获取api的数据，子组件的mounted是拿不到的

五、在Vue中组件常用的通信方式？

- 父向子组件传值
 - 引用子组件时，在子组件身上bind绑定指定数据名
 - 子组件实例属性props里进行声明获取（只读不可修改）
 - 传值数据的类型标注都可
- 子组件向父组件传值
 - 自定义事件函数绑定在子组件身上 `this.$emit('childFun',data);`
 - data为子组件想像父组件传递的值
 - `<child @childFun="parFun"></child>`
 - 父组件中需要定义函数parFun(data)获取
- 兄弟组件间传值
 - 利用eventBus事件总线实现
- 当然，还有更合适的方案是使用vuex来做数据管理。

六、如何简化Vue中重复的东西，比如Vue中（没听清，可能是makething？）怎么使用的

- 这个没回答出来，不知道他说的是啥。

七、Vue中\$nextTick()干嘛使得，应用场景是什么？

- 定义：在下次DOM更新循环结束后执行延迟回调。在修改数据之后立即使用这个方法，获取更新后的DOM。
- 理解：nextTick()是将回调函数延迟在下一次dom更新数据后调用，即当数据更新了在DOM中渲染后自动执行该函数。
- 背景：created时，data和methods初始完毕，但DOM未渲染。
- mounted时，页面的DOM节点才被初始化后挂载。
- 因此在**created**操作DOM，都需要将js代码放进Vue.nextTick()的回调函数中。

八、Vue中有进行过什么性能方面的优化吗？

- 开放性回答：
 - 比如组件for循环时，设置唯一key值，便于vue更快定位该数据
 - 使用keep-alive进行组件缓存
 - 三方库按需引入，避免无用加载等

九、Vue中对于data中定义的数据，但不需要监听怎么处理，比如Object.freeze()

- 定义：Object.freeze() 方法可以冻结一个对象。这个对象不能被修改添加和删除
- 在确定不会操作此对象时，使用Object.freeze()可以让性能大幅提升

感受：Vue方面，更注重生产上的实际使用和场景解决，不过路由和Vuex没问挺意外。

一、常见的请求方法

- GET方法，从指定资源中请求数据。（GET请求有长度限制，可缓存不可修改）
- POST方法，数据传输用于创建或更新资源（POST无长度限制，不可缓存）
- PUT方法，它和post方法很相似，不过请求相同时PUT会覆盖前一个，post不会
- DELETE方法，用来删除指定资源

二、通信时常用的状态码

- 200 OK：表示客户端请求在服务器端正常处理。
- 3XX 重定向：表明浏览器需要执行某些特殊的处理以正确处理请求。
 - 301 Moved Permanently（永久性重定向）
 - 302 Found（临时重定向）
 - 304 Not Modified 允许请求访问资源，但未满足条件的情况。
- 4XX 客户端错误
 - 400 Bad Request：该状态码表示请求报文中存在语法错误
 - 404 Not Found（未找到）表明服务器上无法找到请求的资源
- 5XX 服务器错误
 - 500 Internal Server Error（内部服务器错误）
 - 503 Service Unavailable（暂停服务）

三、HTTP中在头部中怎么设置缓存多久？

- 缓存过期时间，用来指定资源到期的时间，是服务器端的具体的时间点
- Expires=max-age + 请求时间，需要和Last-modified结合使用
- 通常是Cache-Control，相对时间
 - 例如设置 Cache-Control:max-age=31536000
 - 有效期为 (31536000 / 24 / 60 * 60) 天

四、跨域的通常解决方案？

- 这个话题说烂了，不再重复

五、目前主流的HTTP版本？

- 主流的是HTTP 1.0/ HTTP 1.1 / HTTP 2.0(未广泛运用)
- 内容过多，不赘述

六、从浏览器地址栏输入 url 到显示页面的步骤（QAQ，烂大街的问题）

1. 要解析域名转换成对应的公网 IP；
 - 域名解析为 Ip 地址，依靠 DNS 服务，DNS 服务将该域名解析为 IP 地址。如果这一级的 DNS 服务器上找不到，则会进步向上一级的 DNS 发去查询请求。直到根域名服务器为止。如果中间找到了对应的公网 Ip，则再一级一级返回，如果找不到则返回错误信息，域名解析失败。
2. 根据公网 IP 通过庞大的互联网路由到对应的服务器上；
 - 获取公网 IP 地址，就可以通过路由器找到对应的服务器所在。
3. 建立可靠的 TCP 数据连接；
 - 这里使用的 TCP 协议在使用前需要首先在客户端和服务端之间，建立起一条数据链路。分为三步，也叫作 TCP 三次握手。
4. 服务器对该 URL 中的请求进行处理分发，返回一个 html；

- 客户端根据 URL 携带的不同信息，对其进行分发和数据处理。通过对应的 URL 格式 dispatch 到不同的后台处理 Java 类中，进行数据处理后，再返回对应的 HTML 文件给客户端
5. 浏览器或者客户端对该 HTML 进行渲染；
- 网页多是动态 HTML。里面包含了大量的 JS 代码。目前的浏览器为了尽快加载页面，普遍采用了一边解析一边加载的原则

七、浏览器的解析再详细描述一下

- 表述了下获取的HTML和CSS怎么边解析边加载的规则及过程。

感受：HTTP很基础很简单，过程穿插了TCP握手时SYN数据及ACK确认包及对称与非对称加密原理（是否共享秘钥）

JavaScript相关

一、JavaScript大致的两种数据类型，在内存存储上有什么区别？

- 指的就是基本数据类型与引用数据类型
- 基本数据类型
 - 保存在栈内存中，它占据空间小、大小固定，便于频繁使用。
- 引用数据类型
 - 引用数据类型是保存在堆内存中
 - 栈内存中保存一个对堆内存中实际对象的引用（其实就是地址）

感受：啊？然后我内心就震惊了，JS相关的这就没了？作用域、原型链、防抖节流，ES6新语法啥都没问

拓展

一、看你简历上自己也写过PHP，那了解过什么API设计规范吗？比如RESTful API

- 得，给自己挖了一个大坑，就写过增删改查，设计规范这玩意我哪知道
- 含含糊糊随便说了下，最后说自己对这方面不太擅长。
- [阮一峰老师RESTful API 设计规范](#)

二、前端项目一般是怎么部署的？

- 这个我没有在生产上部署过，根据自己的demo回答的很烂
- webpack自动化项目
- 通过npm run build 工程项目生成dist文件
- 打包生成的dist文件移植到Tomcat服务器下

三、工作中Git的使用

- 常用的pull和push的没说
 - 表述常用的checkout，以及区分需求节点如何打tag
 - 涉及代码merge时，dev及 release开发分支区分
 - 冲突如何解决，以及如何选用多个commit的其中几个
 - 我回答平时使用cherry-pick解决
 - 公司的代码如何保管的（内部搭建私有服务器）
- ### 四、拓展内容
- docker linux nginx mock的了解及使用

- 工作中联调是怎么样的，公司API规则
- 平时通过哪些渠道获取技术信息，有哪些学习习惯

想了解的东西？

- 公司目前的项目业务发展方向及技术栈，技术成员情况怎样，工作所在地。

总结

- 岗位性原因，感觉确实是以解决生产实用性为主，原生的JS语法及相关开发技巧问的太少，关注点在于Vue作为工具的开发上，对面试者JS基础问的少，说明公司可能不会太过于关注员工的自我成长。