

# 面试题分享

## 1.在项目中有没有单独封装组件

答：有的，在项目的 `common` 文件下会存放项目公用组件（如：页面的头组件、页面底部组件等）项目里的 `feature` 文件下则是放项目的功能组件（如：轮播图、分页器、上拉加载这些功能组件）把这些页面重复的部分，抽离出来进行单独的封装，有效的减少了代码量，提升了项目的开发效率。解决了传统项目中效率低、难维护、复用性低的等问题。

## 2.在项目中发送请求怎么携带 token

答：当用户登陆的时候，后端会把用户的信息和 token 返回给我们的，我们进行存储 token，把存储的 token，放在请求拦截器中，这样所有的请求都可以直接通过请求拦截器将 token 传给服务器。

## 3.工作中有用到 git 吗

答：有的，在之前的公司里，基本上用的都是 git 进行多人开发，git 是一个分布式版本管理工具，首页是获取项目：访问远程仓库复制项目链接，下载项目到本地。使用 `git init` 方法在项目根目录下，创建本地 git 仓库，进行一个开发，开发完毕后使用 `git add .` 方法将文件都加载到暂存区，再使用 `git commit -m` 提交代码

开发完毕后将文件都加载到暂存区，再使用 `git commit -m` 提交代码到本地仓库，先 `pull` 远程仓库的代码（避免远程仓库与本地代码不一致时发生冲突），最后 `push` 本地仓库的代码提交到远程仓库。还可以使用 `git merge` 将子分支代码合并到主分支代码仓库里。

#### 4.git 版本发生冲突你怎么解决的

答：版本冲突基本出现在合并操作（合并远程仓库代码或者合并分支代码）中。如果出现版本冲突，需要具体分析出现冲突的代码区，手动进行代码合并，然后再进行提交。

#### 5.如何快速让一个盒子水平垂直居中

纯 flex 布局实现：给元素设置 flex 布局，添加 flex 的 `alignItems:center` 垂直居中，`justifyContent:center` 水平居中，实现居中效果；  
flex 布局 + 外边距 (margin) 实现：给父级设置为 `display: flex;` 布局，/ 给子级添加 `margin:auto;` 就可以实现水平垂直居中显示；  
绝对定位 + 外边距 (margin) 实现：绝对定位的元素将上下左右都设置为 0，再设置 `margin:auto` 即可实现居中；  
绝对定位 + 外边距 (margin) + C3 的转换 transform 中的 `translate` 实现：绝对定位元素，利用 margin 偏移外容器的 50%，再利用 `translate` 回补自身宽高的 50%；

## 6.首屏加载白屏怎么进行优化

答：

1. 使用CDN 减小代码体积，加快请求速度；
2. SSR 通过服务端把所有数据全部渲染完成再返回给客户端；
3. 路由懒加载，当用户访问的时候，再加载相应模块；
4. 使用外链CSS，JS文件；
5. 开启 GZIP 压缩；(是一种http请求优化方式，通过减少文件体积来提高加载速度。html、js、css文件甚至json数据都可以用它压缩，可以减小60%以上的体积。(需要后端支持))
6. 项目打包不生成 .map 文件；
7. 使用骨架屏 在首页加载时，显示页面结构轮廓；

## 7.路由传参 query与 params 区别

答：query类似 get，跳转之后在地址栏中显示拼接参数，类似?id=1；params类似 post，跳转之后不在 url 中显示参数；相对于query比较安全；

注意点：(params在进行路由传参时，要在路由中配置占位。如果不在路由中配置占位，跳转过去以后刷新页面会数据丢失，query参数数据不会丢失)；

由中配置占位。如果不在路由中配置占位，跳转过去以后刷新页面会数据丢失，query 参数数据不会丢失)；

## 8.你是怎么理解 Vuex 的

答：是 Vue 开发的状态管理模式。Vuex 就相当于是一个大仓库用来存储数据，可以实现组件间数据共享；也是组件间通信的一种，适用于任意组件通信；

### Vuex 有五种属性

- state：仓库存储响应式数据的地方；
- mutation：修改 state 中的数据唯一途径；
- action：输写业务逻辑，处理异步操作；
- getters：理解为计算属性，用于简化仓库数据，让组件获取仓库的数据更加方便；
- modules：用于实现 Vuex 模块化；

## 9.项目基础优化

答：

1. 减少 HTTP 请求数 (避免重复请求)
2. 减少 DOM 元素和 DOM 操作
3. 使用雪碧图
4. 压缩合并 JS、CSS 代码
5. 避免图片 src 为空
6. 把样式表放在 link 中
7. css 样式放置在文件头部
8. js 脚本放置在文件末尾

## 10.对 \$nextTick 异步渲染对理解

答：Vue 采用了数据驱动视图的思想，但是在一些情况下，仍然需要操作 DOM。有时候，DOM1 的数据发生了变化，而 DOM2 需要从 DOM1 中获取数据，那这时就会发现 DOM2 的视图并没有更新，这时就需要用到了 nextTick 了；

**作用：**在下一次 DOM 更新结束后执行其指定的回调；(当数据更新后，要基于更新后的新 DOM 进行某些操作时，要在 nextTick 指定的回调函数中执行)。如果不采用异步更新，在每次数据的更新后，都会对当前组件进行重新渲染。所以为了性能考虑，vue 会在本轮数据更新后，再去异步更新视图；