## Vue权限控制

<https://github.com/febobo/web-interview/issues/29>

### 接口权限:

请求时带上token（header）。

没有请求接口的权限（token是否有效）时，后端返回协定code，前端识别后跳转至登录页。

### 路由权限（跳转）:

1. 一次性挂载全部的前端路由表，路由表中使用meta的roles中添加路由的权限信息，在全局路由守卫中进行权限判断。
2. 初始只加载不需要权限判断的路由，获取用户权限信息后，筛选出合适的路由再添加至路由表。

### 菜单权限:

1. 菜单数据由后端做好权限处理然后返回，菜单name与前端路由name相匹配，加载路由后全局路由守卫中进行判断。
2. 菜单数据和路由表都由后端返回（一个数据源），组件component用字符串表示，前端替换成vue路由组件。

### 按钮权限：

1. 页面内获取用户权限role和路由表里的meta.btnPermissions，v-if控制按钮显示。
2. 通过自定义指令。路由表中使用meta.btnPermissions定义权限信息，然后自定义一个指令，指令根据路由信息决定宿主元素是否显示。

## keep-alive原理

三个属性：

include（缓存白名单）、exclude（缓存白名单）、max（最大缓存数）。

存储两个对象：

this.cache（存储虚拟dom）和this.keys（存储虚拟dom的key，存放虚拟dom顺序，超出max后把最不常用的删除掉）。

keep-alive的渲染，keep-alive并不会生成新的dom节点：

组件的配配置项abstract为true，则这个组件不会渲染成真正的dom。具体做法是在初始化生命周期的时候，会对组件对象的$parent设置为第一个不为abstract的父组件（往上找）。

如何利用缓存实现组件（缓存）：

第一次访问keep-alive组件时，会把子组件的虚拟dom添加到缓存中，然后直接渲染子组件。

第二次访问keep-alive组件时会把从缓存里取出这个子组件虚拟dom对象，把它赋值给子组件的vnode.componentInstance，在update => patch的时候进行判断，如果存在vnode.componentInstance，直接取vnode.elm（上次渲染的真实dom）进行渲染。

keep-alive生命周期：

一般的组件，每一次加载都会有完整的生命周期，即生命周期里面对应的钩子函数都会被触发；而keep-alive包裹组件中，第一次进入的触发顺序为created=>mounted=>activated，再次进入只会触发 activated，退出时触deactivated。

当vnode.componentInstance和keepAlive同时为truly值时，不再进入$mount过程，那mounted之前的所有钩子函数（beforeCreate、created、mounted）都不再执行。

当vnode.keepAlive为true时，会在update => patch的时候递归地去执行所有子组件的activated钩子函数。

## **ref与reactive的区别**

一般reactive是用来定义响应式（对象）类型的数据，reactive 是引用类型的。

ref是定义响应式普通类型的数据，ref是基本类型的，ref 修改值时，需要在变量名后加 .vlaue 才能修改值，内部是使用get set 来操作，reactive 则是 Proxy实现。