技术难点重点

1. 开发语言：Java

2. 数据库：MySQL，hutool的加密模块

3. 前端：Vue.js、axios，JavaScript等

4. 后端：Spring Boot，mybatis，maven，hutool，knife4j，postman

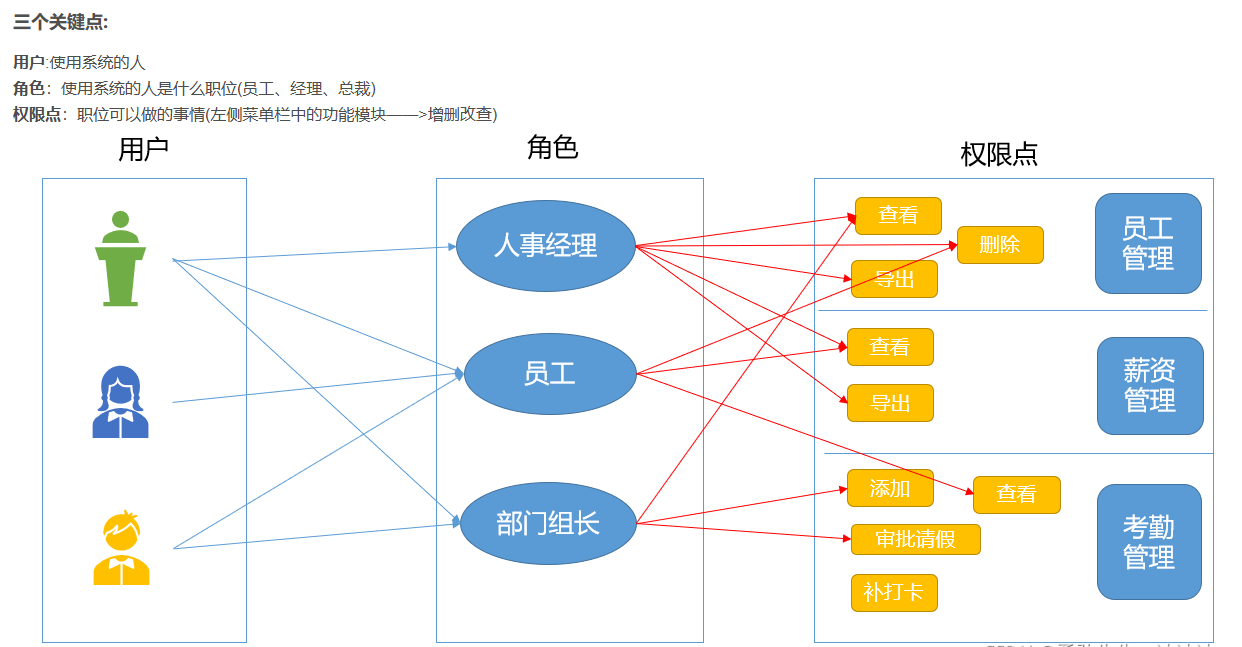
管理系统实现权限控制机制，在项目中使用sa-token框架来实现权限控制机制

1. 如何设计数据库以建立用户-角色-菜单之间的联系
2. 如何查询与处理树结构的数据
3. Vue 如何实现动态加载路由

<http://t.csdn.cn/ddiqa>

数据库设计在数据库中设计权限认证，通常可以采用RBAC（Role-Based Access Control）模型。

|  |  |  |
| --- | --- | --- |
| 用户 | 角色 | 权限 |
| Admin | 超级管理员 | 所有权限 |
| Teacher | 普通管理员 | 本用户和student权限 |
| Student | 普通用户 | Student权限 |



动态权限修改，采用RBAC模型，前端菜单和后台权限实时更新是怎么做到的？

实现动态权限修改通常需要以下几个步骤：

建立前端路由映射表和后台权限表

前端路由映射表用于记录每个页面的URL路径和名称，后台权限表用于记录每个功能操作所需的权限。

根据用户角色获取权限列表

当用户登录系统时，根据用户角色查询其所拥有的权限列表，并将其存储在缓存或客户端存储中。

在后台修改权限

当管理员修改用户权限时，后台需要实时更新数据库中的权限表。同时，也要清除缓存或客户端存储中与该用户相关的权限信息，以便下次访问时重新获取最新的权限列表。

实现前端菜单和按钮的动态显示和隐藏

前端可以根据当前用户角色和权限列表动态生成菜单和按钮，并根据权限控制其显示和隐藏。例如，在Vue框架中，可以使用v-if指令来控制元素的显示和隐藏。

实现页面访问控制

在前端路由配置中可以设置页面的访问权限，只有当用户拥有相应的权限时才能访问该页面。在Vue框架中，可以使用导航守卫来实现页面访问控制。

通过以上步骤，就可以实现动态权限修改，并且前端菜单和后台权限实时更新的功能。

校园管理系统的学生信息管理模块数据库设计，包括ER图，实体类，创建数据库的sql文件

Spring Boot作为后台接口，Vue作为前端框架实现动态生成菜单路由和权限控制

定义了一个MenuController用于返回当前登录用户所拥有的菜单数据。在前端页面中使用Vue Router来配置路由表，并根据返回的菜单数据动态渲染左侧菜单栏。同时，通过判断当前路由是否与菜单项对应，实现了简单的权限控制。

一般的校园管理系统中，各个用户对系统资源的访问权限是通过RBAC模型来实现的，即基于角色的访问控制模型。具体来说，系统中会定义多个角色，每个角色会被赋予不同的权限，而用户则会被分配到相应的角色中，从而获取相应的权限。

在数据库的设计方面，可以考虑以下表格：

1. 用户表格：用于存储系统中的用户信息，例如用户ID、用户名、密码等等。
2. 角色表格：用于存储系统中的角色信息，例如角色ID、角色名称、角色描述等等。
3. 权限表格：用于存储系统中的权限信息，例如权限ID、权限名称、权限描述等等。
4. 用户角色关联表格：用于记录用户和角色之间的关联信息，例如用户ID和角色ID。
5. 角色权限关联表格：用于记录角色和权限之间的关联信息，例如角色ID和权限ID。
6. 其中，用户表格和权限表格是系统中必须的表格，用户角色关联表格和角色权限关联表格都是用于关联用户、角色和权限之间关系的。在设计关联表格时，可以使用外键来关联相关表格。

这种设计方式可以使系统中的权限管理更加规范和灵活，可以有效避免一些安全漏洞的出现。但是，具体的数据库设计仍然需要根据具体的系统需求来进行合理的设计。

在数据库设计完成后，需要在系统代码中实现RBAC模型的功能。具体步骤如下：

1. 定义角色及其权限：在系统代码中定义角色及其权限，例如管理员角色可以拥有对学生信息、教师信息等资源的管理权限，而普通用户则只能访问这些资源。
2. 分配角色给用户：在用户管理模块中，允许系统管理员将角色分配给具体的用户，并将该信息保存到用户角色关联表格中。
3. 验证用户权限：在系统各个模块中，需要对用户进行身份验证，并验证其是否拥有对某项资源的访问权限。如果没有权限，则需要提示用户或者限制其操作。
4. 控制用户权限：在系统中，需要对用户权限进行控制，例如限制用户访问频率、记录用户操作日志等等。

在实现这些功能时，可以根据具体的编程语言和开发框架选择相应的权限控制库或工具，例如在Java开发中可以使用Spring Security框架来实现RBAC模型的功能。

问题：

角色和权限集合关联添加外键

举例在vue添加请求拦截器，并携带content-type，cookie

// 请求拦截器

instance.interceptors.request.use(

(config) => {

// 在请求头中添加 Content-Type 和 Cookie

config.headers['Content-Type'] = 'application/json';

config.headers['Cookie'] = `sessionId=${localStorage.getItem('sessionId')}`;

return config;

},

(error) => {

return Promise.reject(error);

}

);

但是上面方法cookie不能跨域

在 Vue 中，我编写了 Axios 拦截器来添加请求拦截器，并携带 Content-Type 和 Cookie，为什么请求接口的时候没有携带cookie？

1. 在 Vue 中添加请求拦截器并设置 Cookie，需要注意以下几点：
2. 在携带 Cookie 时，需要确保 Cookie 对应的域名和当前请求的域名一致。否则，浏览器会将请求中的 Cookie 视为无效，并忽略其携带。
3. 如果域名一致，但是 Cookie 的 path 属性不匹配当前请求的路径，则同样会被视为无效。
4. 如果仍然无法成功携带 Cookie，可以考虑将 withCredentials 属性设置为 true，用于指示在跨域请求时是否携带身份凭证（例如 Cookie）。

://localhost:8081/notice/selectAll' from origin 'http://localhost:8080' has been blocked by CORS policy: Response to preflight request doesn't pass access control check: The value of the 'Access-Control-Allow-Origin' header in the response must not be the wildcard '\*' when the request's credentials mode is 'include'. The credentials mode of requests initiated by the XMLHttpRequest is controlled by the withCredentials attribute.是什么意思，哪里出了问题？

这是一个跨域资源共享（CORS）的错误提示。当你在前端向不同域名的后端（例如从 localhost:8080 向 localhost:8081 发送请求）发出带有身份凭证（如 Cookie）的 AJAX 请求时，浏览器会首先执行一个预检请求（preflight request）——发送一个 OPTIONS 请求，以便确保服务器可以接受这个请求。

vue怎么将后端返回的权限码集合保存在localStorage，举个例子

// 后端返回的权限码 List 集合

const permissionList = ["permission1", "permission2", "permission3"];

// 将权限码 List 集合保存到 localStorage 中，键为 "permissions"

localStorage.setItem("permissions", JSON.stringify(permissionList));

// 从 localStorage 中获取保存的权限码 List 集合

const permissions = JSON.parse(localStorage.getItem("permissions"));

// 在 Vue data 中保存权限码 List 集合

data() {

return {

permissions: permissions

}

}

我们首先将后端返回的权限码 List 集合保存到 localStorage 中，使用 JSON.stringify() 方法将其转换为 JSON 格式的字符串。键名为 "permissions"，可以根据实际需求进行修改。

接下来，在需要使用权限码 List 的地方，通过 JSON.parse() 方法从 localStorage 中获取保存的权限码 List 集合，并赋值给变量 permissions。

如果你使用了 Vue，也可以在 Vue data 中保存权限码 List 集合，以便于在组件中使用。例如，将获取到的 permissions 存储在 data() 中，然后在组件中通过访问 this.permissions 来获取权限码 List。

需要注意的是，localStorage 存储的数据类型只能是字符串，因此需要使用 JSON.stringify() 和 JSON.parse() 方法进行转换。而在使用 localStorage 时，还需要注意数据类型的一致性和值的有效性。

无论是app还是小程序，其传递方式都大同小异。

那就是，将 token 塞到请求header里 ，格式为：{tokenName: tokenValue}。

以经典跨端框架 vue为例

// 添加请求拦截器

service.interceptors.request.use((config) => {

config.headers['satoken'] = getToken('satoken')

return config

}, (error) => {

return Promise.reject(error)

})

export function setPermission(perListOne,permission) {

return localStorage.setItem(perListOne,permission)

}

export function getPermission(perListOne) {

return localStorage.getItem(perListOne)

}

login(this.form).then(res => {

if (res.data.code === 200) {

setToken('username', res.data.data.roleList)

setToken('satoken', res.data.data.token)

setPermission('permissionList', res.data.data.permissionList)

this.$message({message: res.data.message, type: 'success'})

this.$router.push('/home')

}

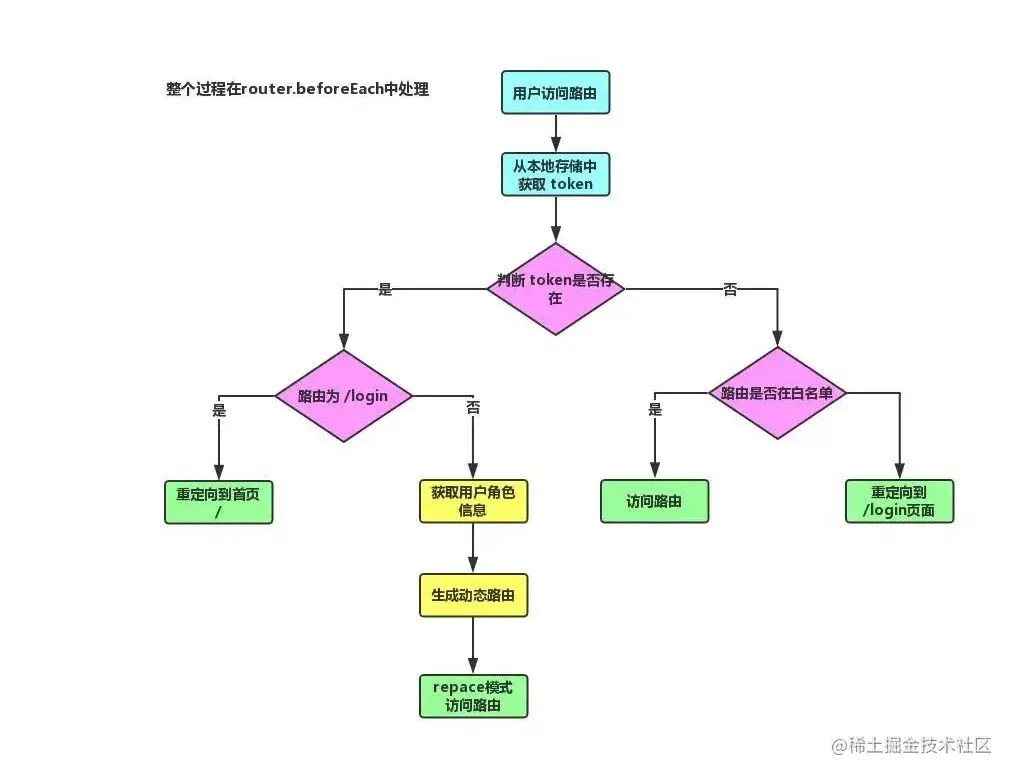
})

vue获取到了后端返回的权限码，怎么通过权限码来显示vue页面或者通过权限码精确显示每个按钮？

对于权限控制，我们可以通过前端路由和动态组件的方式实现页面级别的权限控制，通过指令或者自定义组件来实现按钮级别的权限控制。

vue获取到了后端返回的权限码存在localstage，怎么实现路由分为两部分,一部分是固定的几个，另一部分是根据权限动态生成，目的隐藏掉不具有权限的菜单(路由)和按钮，具体怎么实现？

目前开发到可以精确控制显示按钮，接下来管理员要开发功能控制学生的权限，教师的权限，用户管理



RBAC模型：用户拥有指定的角色，角色再拥有指定的权限

采用 [账号id -> 角色id -> 权限列表] 的缓存模型时，则只需要清除或修改 [角色id -> 权限列表] 一条缓存即可