后端权限处理

权限设计

数据表关系

- 角色（group）与资源（menu、element）关联（base_resource_authority）
- 用户（user）与角色（group）关联（group_leader、group_memeber）

前后端资源映射关系

- 菜单（menu）`code` 对应前端路由的`authority`
- 资源（element）`code` 对应页面的按钮条件
- 资源（element）`uri` 、`code` 对应配置网关拦截的Url

用户权限加载

类`PermissionService` 中的获取用户权限方法`getPermissionByUsername`

```
public List<PermissionInfo> getPermissionByUsername(String username) {
        User user = userBiz.getUserByUsername(username);
        List<Menu> menus =
menuBiz.getUserAuthorityMenuByUserId(user.getId());
        List<PermissionInfo> result = new ArrayList<PermissionInfo>();
        PermissionInfo info = null;
        menu2permission(menus, result);
        List<Element> elements =
elementBiz.getAuthorityElementByUserId(user.getId() + "");
        element2permission(result, elements);
        return result;
    }
```

权限拦截

- 网关核心权限拦截类`AccessGatewayFilter`

```
@Override
    public Mono<Void> filter(ServerWebExchange serverWebExchange,
GatewayFilterChain gatewayFilterChain) {
        log.info("check token and user permission....");
        LinkedHashSet requiredAttribute =
serverWebExchange.getRequiredAttribute(ServerWebExchangeUtils.GATEWAY_OR
IGINAL_REQUEST_URL_ATTR);
        ServerHttpRequest request = serverWebExchange.getRequest();
        String requestUri =
request.getPath().pathWithinApplication().value();
        if (requiredAttribute != null) {
            Iterator<URI> iterator = requiredAttribute.iterator();
            while (iterator.hasNext()) {
                URI next = iterator.next();
                if (next.getPath().startsWith(GATE_WAY_PREFIX)) {
                    requestUri = next.getPath();
                }
            }
        }
        final String method = request.getMethod().toString();
        ServerHttpRequest.Builder mutate = request.mutate();
        // 不进行拦截的地址
        if (isStartWith(requestUri)) {
            ServerHttpRequest build = mutate.build();
            return
gatewayFilterChain.filter(serverWebExchange.mutate().request(build).buil
d());
        }
        UserInfo userInfo;
        try {
            userInfo = getJWTUser(request, mutate);
        } catch (Exception e) {
            log.error("用户Token过期异常", e);
            return getVoidMono(serverWebExchange, new
TokenForbiddenResponse("User Token Forbidden or Expired!"));
        }
        List<PermissionInfo> permissionIfs =
userService.getAllPermissionInfo();
        // 判断资源是否启用权限约束
        Stream<PermissionInfo> stream = getPermissionIfs(requestUri,
method, permissionIfs);
        List<PermissionInfo> result =
stream.collect(Collectors.toList());
        PermissionInfo[] permissions = result.toArray(new
PermissionInfo[]{});
        if (permissions.length > 0) {
            if (checkUserPermission(permissions, serverWebExchange,
userInfo)) {
                return getVoidMono(serverWebExchange, new
TokenForbiddenResponse("User Forbidden!Does not has Permission!"));
            }
        }

        ServerHttpRequest build = mutate.build();
        return
gatewayFilterChain.filter(serverWebExchange.mutate().request(build).buil
d());

    }
```

前端权限处理

在用户登录成功之后才会进行菜单、页面资源权限的加载。

路由与菜单控制

- 动态权限路由控制，`src/store/modules/permission.js` ，`GenerateRoutes` 该方法在`AG-Admin-UI/src/main.js` 中被调用到

```
GenerateRoutes({
      commit
    }, menus) {
      return new Promise(resolve => {
        getAllMenus().then(data => {
          const menuDatas = {};
          for (let i = 0; i < data.length; i++) {
            menuDatas[data[i].code] = data[i];
          }
          // 此处根据当前用户的菜单访问权限来动态过滤
          const accessedRouters = filterAsyncRouter(asyncRouterMap,
menus, menuDatas);
          console.log(accessedRouters);
          commit('SET_ROUTERS', accessedRouters);
          resolve();
        });
      })
    }
```

- 菜单生成过程，`src/views/layout/Sidebar.vue` 、`src/views/layout/SidebarItem.vue` ，利用递归来生成菜单，建议菜单的层级控制到2-3层。

```
<template v-for="item in routes">
                <el-submenu :index="item.title" :key="item.name">
                        <template slot="title">
                                <icon-svg v-if='item.icon' :icon-
class="item.icon"></icon-svg>
                                <span>{{item.title}}</span>
                        </template>
                        <template v-for="child in item.children">
                                <sidebar-item class='nest-menu' v-
if='child.children&&child.children.length>0' :routes='[child]'
:key="child.name"> </sidebar-item>
                                <a target="_blank"  v-
if="child.href!=null&&child.href.indexOf('http')==0" :href="child.href"
:key="child.name">
                                        <el-menu-item
:index="'/'+item.code+'/'+child.code">
                                                <icon-svg v-
if='child.icon' :icon-class="child.icon"></icon-svg>
                                                <span>{{child.title}}
</span>
                                        </el-menu-item>
                                </a>
                                <router-link v-
if="child.href!=null&&child.href.indexOf('http')!=0&&child.type!='dirt'"
:to="'/'+item.code+'/'+child.code" :key="child.name">
                                        <el-menu-item
:index="'/'+item.code+'/'+child.code">
                                                <icon-svg v-
if='child.icon' :icon-class="child.icon"></icon-svg>
                                                <span>{{child.title}}
</span>
                                        </el-menu-item>
                                </router-link>
                        </template>
                </el-submenu>
        </template>
```

页面按钮权限控制

- 加载用户页面权限资源，`src/store/modules/user.js` 的`GetInfo` 该方法在`src/main.js` 中被调用到

```
    // 获取用户信息
    GetInfo({
      commit,
      state
    }) {
      return new Promise((resolve, reject) => {
        getInfo(state.token).then(response => {
          const data = response;
          commit('SET_ROLES', 'admin');
          commit('SET_NAME', data.name);
          commit('SET_AVATAR',
'http://git.oschina.net/uploads/42/547642_geek_qi.png?1499487420');
          commit('SET_INTRODUCTION', data.description);
          const menus = {};
          for (let i = 0; i < data.menus.length; i++) {
            menus[data.menus[i].code] = true;
          }
          commit('SET_MENUS', menus);
          const elements = {};
          for (let i = 0; i < data.elements.length; i++) {
            elements[data.elements[i].code] = true;
          }
          // 加载按钮资源权限编码
          commit('SET_ELEMENTS', elements);
          resolve(response);
        }).catch(error => {
          reject(error);
        });
        getMenus(state.token).then(response => {
          console.log(response)
          commit('SET_PERMISSION_MENUS', response);
        });
      });
    },
```

- 页面权限控制，示例页面（用户管理），`src/views/admin/user/index.vue`

```
<!-- v-if控制按钮的隐藏和显示 -->
    <el-button class="filter-item" v-if="userManager_btn_add"
style="margin-left: 10px;" @click="handleCreate" type="primary"
icon="edit">添加</el-button>
```

```
// 加载用户
import { mapGetters } from 'vuex';
export default {
 name: 'user',
  data() {
    return {
      userManager_btn_add: false,

    }
  },
    created() {
        // 此处值与上述按钮的v-if变量一致
     this.userManager_btn_add = this.elements['userManager:btn_add'];
  },
  computed: {
    ...mapGetters([
      'elements'
    ])
  },
}
```