

系统组件文档

产品版本：V1.0

文档版本：202401

北京有生博大软件股份有限公司

目录

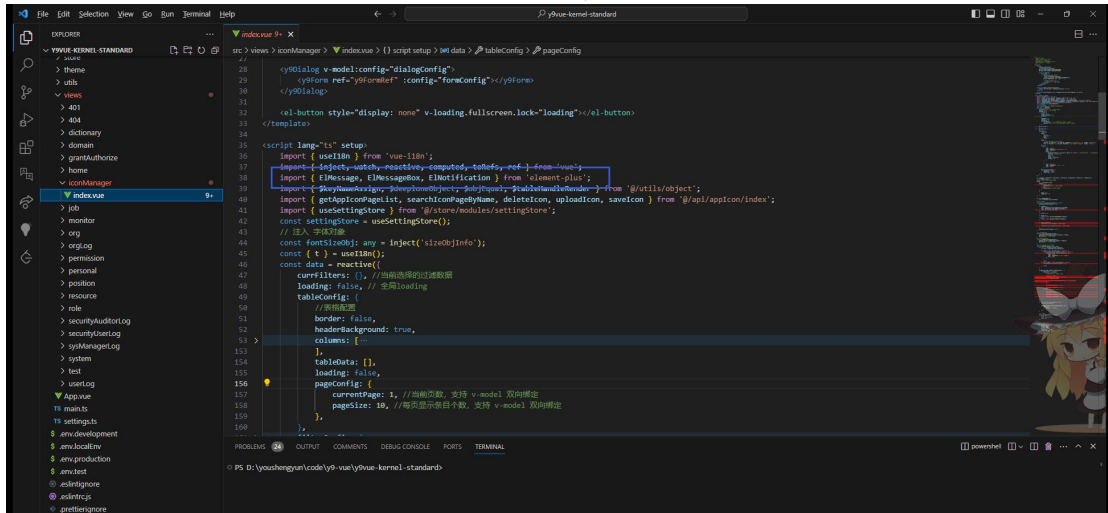
| | |
|-----------------|---|
| 1. 第三方组件 | 3 |
| 2. 内部组件 | 4 |
| 3. 组件自动引入 | 6 |
| 4. 插件组件 | 9 |

1. 第三方组件

项目中使用的 UI 组件借助的是 `element-plus`，相关中文网链接为：<https://element-plus.gitee.io/zh-CN/component/button.html>。

1.1. 按需引入

项目目前的组件注册机制是按需注册，在需要用到的页面才引入。



1.2. 全局引入

如果觉得按需引入太麻烦，可以进行全局注册。

```
// main.ts
import { createApp } from 'vue'
import ElementPlus from 'element-plus'
import 'element-plus/dist/index.css'
import App from './App.vue'

const app = createApp(App)

app.use(ElementPlus)
app.mount('#app')
```

2. 内部组件

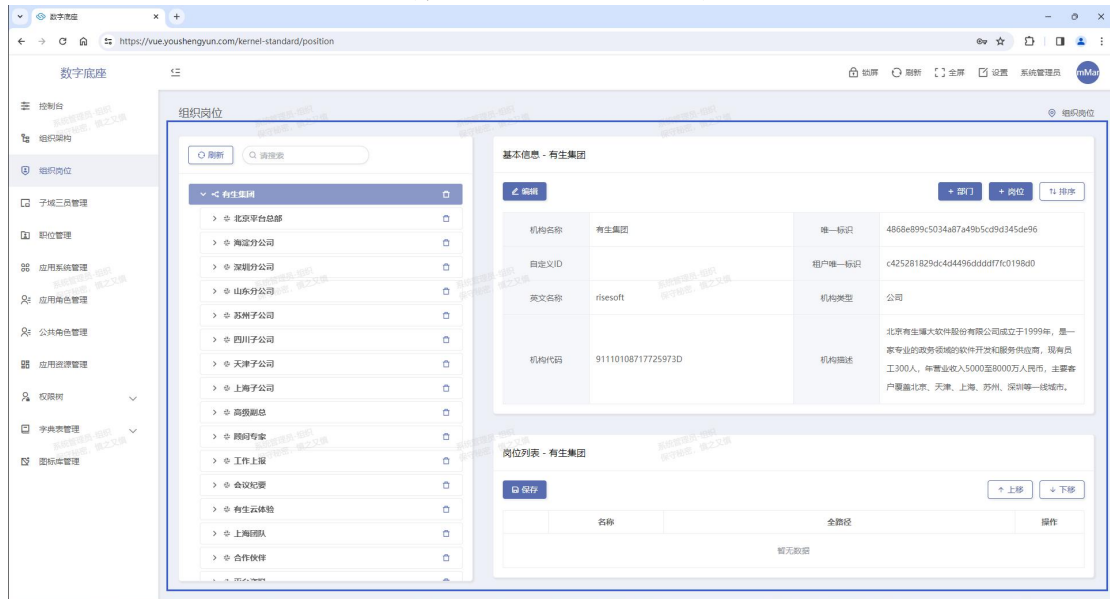
对第三方组件进行再次封装或组合封装。

2.1. 页面组件

组件位置：**src/components/pageModule/fixdTreeModule.vue**

基于树组件和 card 组件进行封装。树的头部，以及右边内容等都使用了插槽，支持内容自定义，提高组件的适用性和稳定性。

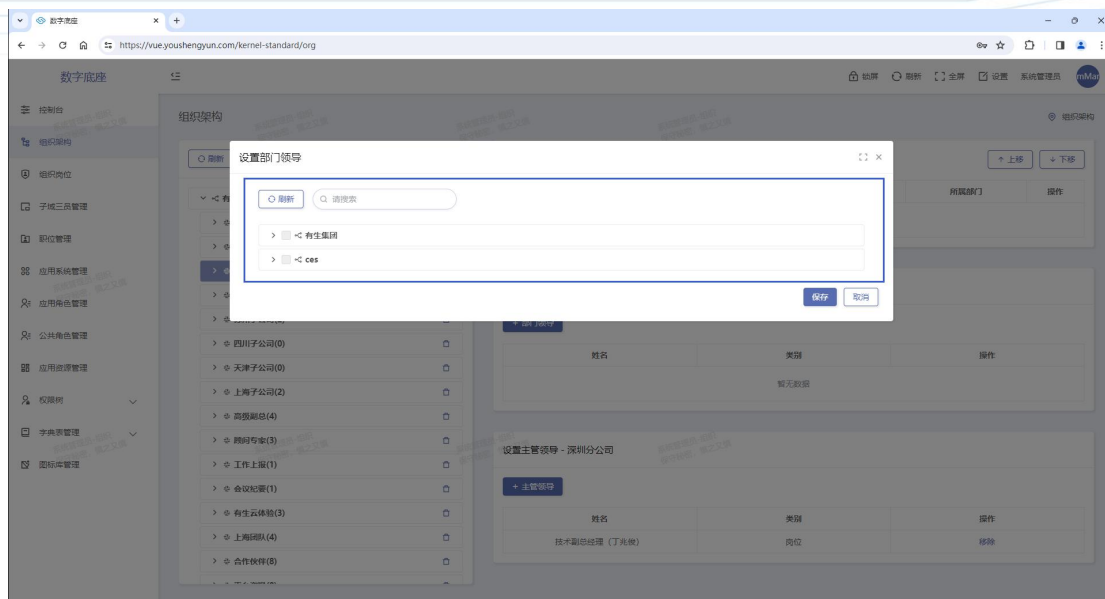
组件内容如下：左边为树，右边为当前点击树数据的相关内容信息。



2.2. 选择组件

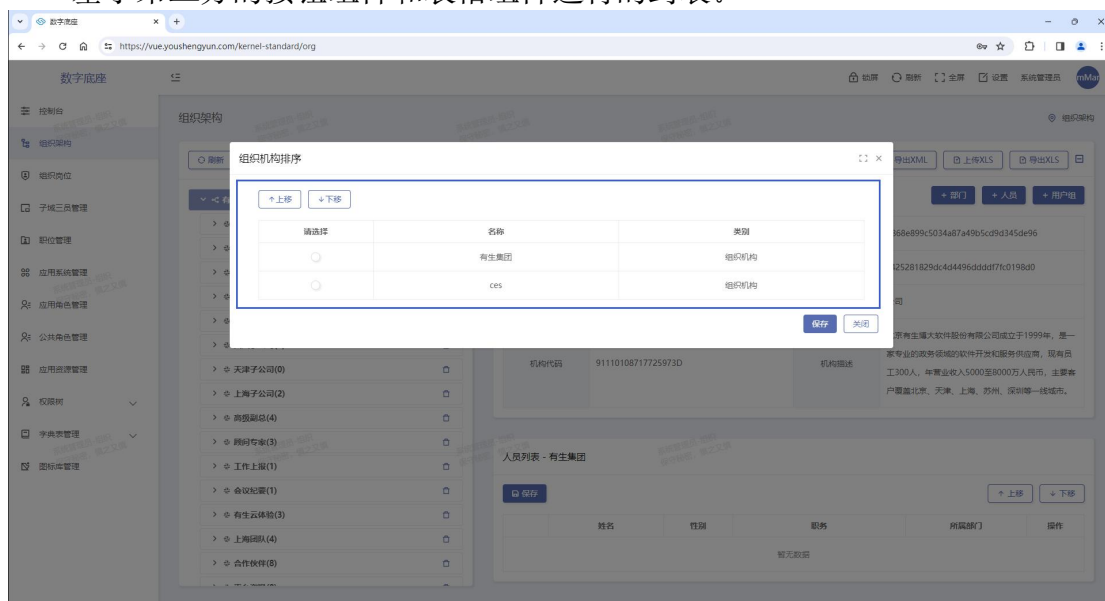
组件位置：**src/components/pageModule/selectTree.vue**

基于第三方的树组件和输入框等组件进行封装。对于树的复选框可设置可选或不可选。



2.3. 排序组件

组件位置: `src/components/pageModule/treeSort.vue`
 基于第三方的按钮组件和表格组件进行的封装。



3. 组件自动引入

为了方便在多个文件内引入多次自定义组件，可以设置组件的自动引入。具体如下：

3.1. 安装插件

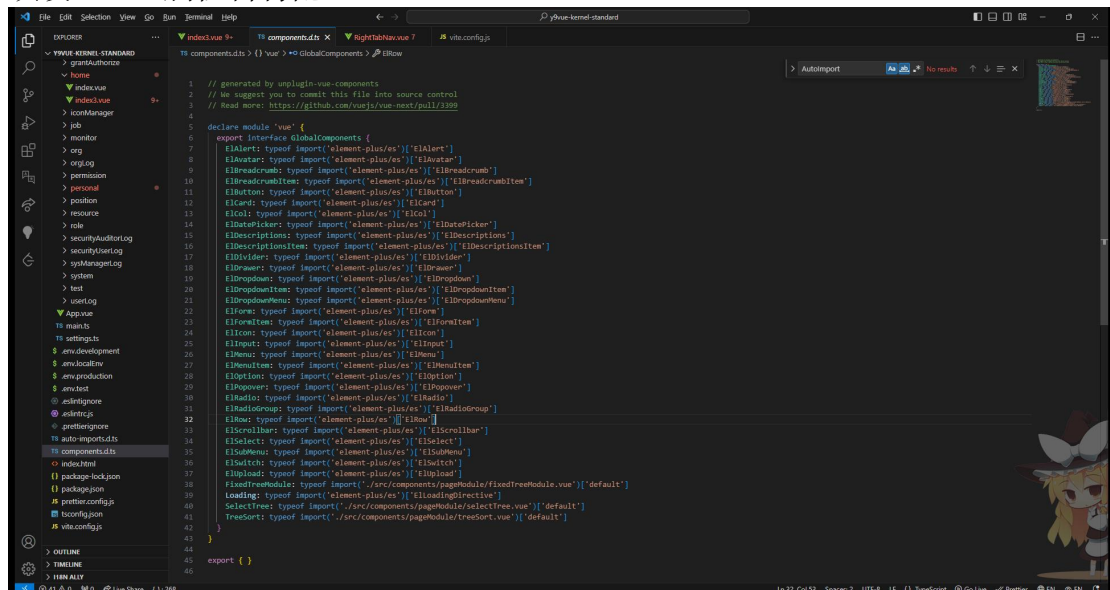
`unplugin-vue-components` 是我们按需引入的核心，可以让我们直接写组件名，就可以在编译时自动添加引入对应组件的代码。

```
npm install unplugin-vue-components -D
```

3.2. 自动引入 element-plus 组件

```
// vite.config.ts
import { defineConfig } from 'vite';
import Components from 'unplugin-vue-components/vite';
import { ElementPlusResolver } from 'unplugin-vue-components/resolvers';
export default defineConfig({
  plugins: [
    Components({
      // ui 库解析器，也可以自定义
      resolvers: [ ElementPlusResolver(),
      // 配置 type 文件生成位置
      dts: 'src/components.d.ts'
    })
  ]
})
```

插件会生成 `components.d.ts`，内容为项目需引入的组件。具体路径可以设置 `dts` 属性自行配置。



3.3. 自动导入自己的组件

vite 配置详情:

```
// vite.config.ts
import { defineConfig } from 'vite';
import Components from 'unplugin-vue-components/vite';
import { ElementPlusResolver } from 'unplugin-vue-components/resolvers';
export default defineConfig({
  plugins: [
    Components({
      // 指定组件所在文件夹的位置，默认是 src/components
      dirs: ['src/components'],
      // ui 库解析器
      resolvers: [ElementPlusResolver()],
      // 配置 type 文件生成位置
      dts: 'src/components.d.ts'
    })
  ]
})
```

自己的组件类型也会生成到 `components.d.ts` 中，在使用组件的文件中，直接写组件名即可，插件会帮你引入进来。

3.4. 部分配置默认说明

下面显示的是配置默认值

```
Components({
  // 用于搜索组件的目录的相对路径。
  dirs: ['src/components'],
  // 组件的有效文件扩展名。
  extensions: ['vue'],
  // 搜索子目录
  deep: true,
  // 自定义组件的解析器
  resolvers: [],
  // 生成`components.d.ts`的全局声明，
  // 也接受自定义文件名的路径
  // 如果安装了 typescript 包 default: true
  dts: false,
  // 允许子目录作为组件的名称空间前缀。
  directoryAsNamespace: false,
  // 忽略名称空间前缀的子目录路径
  // 当`directoryAsNamespace: true`时工作
  globalNamespaces: [],
  // 自动导入指令
  // 默认:'true'用于 Vue 3, 'false'用于 Vue 2
```

```
// Vue 2 需要使用 Babel 进行转换，出于性能考虑，它在默认情况下是禁用的
// 要安装 Babel，运行:' npm install -D @babel/parser '
directives: true,
// 允许组件覆盖具有相同名称的其他组件
allowOverrides: false,
})
```


4. 插件组件

`y9plugin-components` 插件是在 `vue3` 和 `element-plus` 的基础上封装的，具体内容有表单、弹框、表格、树、分页等组件。下载插件，就可以使用该插件内的所有组件效果。说明文档链接为：<https://dev.youshengyun.com/y9vue-code/intro>，查看项目中的前端框架即可。