link: null
title: 珠峰架构师成长计划
description: 可以快速识别.vue文件封装组件插件等功能
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=43 sentences=55, words=572

快速原型开发

可以快速识别.vue文件封装组件插件等功能

```
sudo npm install @vue/cli -g
sudo npm install -g @vue/cli-service-global
vue serve Home.vue
```

关闭eslit提示

```
module.exports = {
    devServer: {
      overlay: {
        warnings: false,
        errors: false
      },
    }
}
```

实现递归菜单组件

```
<menu>
    <menuitem v-for="(item,key) in menuList" :key="key">
        {{item.title}}
    </menuitem>
    <submenu>
        <template #title="title"> &#x6807;&#x9898;1</template>
        <menuitem>&#x6807;&#x9898;1-1</menuitem>
        <menuitem>&#x6807;&#x9898;1-2</menuitem>
            <submenu>
            <template #title="title">&#x6807;&#x9898;1-1-1</template>
            <menuitem>&#x6807;&#x9898;1-1-1-1</menuitem>
            <menuitem>&#x6807;&#x9898;1-1-1-1</menuitem>
        </submenu>
    </submenu>
</menu>
```

根据数据递归渲染

```
menuList:[
    {
        title:'&#x6807;&#x9898;1',
        children:[
            {title:'&#x6807;&#x9898;1-1'},
            {title:'&#x6807;&#x9898;1-2'}
        ]
    },
    {
        title:'&#x6807;&#x9898;2',
        children:[
            {title:'&#x6807;&#x9898;2-1'},
            {title:'&#x6807;&#x9898;2-2'}
        ]
    },
    {
        title:'&#x6807;&#x9898;3',
        children:[
            {title:'&#x6807;&#x9898;3-1'},
            {title:'&#x6807;&#x9898;3-2'}
        ]
    }
]
```

ReSub组件实现

```
<menu>
    <template v-for="(item,key) in menuList">
        <menuitem v-if="!item.children" :key="key">
            {{item.title}}
        </menuitem>
        <resub v-else :key="key" :data="item"></resub>
    </template>
</menu>

// ReSub&#x7EC4;&#x4EF6;&#x4E3B;&#x8981;&#x4F5C;&#x7528;&#x662F;&#x9012;&#x5F52;
<template>
    <submenu class="sub">
        <template #title="title">{{data.title}}</template>
        <template v-for="d in data.children">
            <menuitem v-if="!d.children" :key="d.title">{{d.title}}</menuitem>
            <resub v-else :data="d" :key="d.title" class="sub"></resub>
        </template>
    </submenu>
</template>

<script>
[illegible]
</script>
```

使用vue-cli3.0创建vue项目

```
vue create <project-name>
</project-name>
```

可以通过vue ui创建项目 / 管理项目依赖

```
vue ui
```

配置vue-config.js

```
let path = require('path')
module.exports = {
    publicPath:process.env.NODE_ENV === 'production'? '/vue-project':'/',
    outputDir:'myassets', // &#x8F93;&#x51FA;&#x8DEF;&#x5F84;
    assetsDir:'static', // &#x751F;&#x6210;&#x9759;&#x6001;&#x76EE;&#x5F55;&#x7684;&#x6587;&#x4EF6;&#x5939;
    runtimeCompiler: true, // &#x662F;&#x5426;&#x53EF;&#x4EE5;&#x4F7F;&#x7528;template&#x6A21;&#x677F;
    parallel:require('os').cpus().length > 1, //&#x591A;&#x4F59;1&#x6838;cpu&#x65F6; &#x542F;&#x52A8;&#x5E76;&#x884C;&#x538B;&#x7F29;
    productionSourceMap:false, //&#x751F;&#x4EA7;&#x73AF;&#x5883;&#x4E0B; &#x4E0D;&#x4F7F;&#x7528;soruceMap

    // https://github.com/neutrinojs/webpack-chain
    chainWebpack:config=>{
        // &#x63A7;&#x5236;webpack&#x5185;&#x90E8;&#x914D;&#x7F6E;
        config.resolve.alias.set('component',path.resolve(__dirname,'src/components'));
    },
    // https://github.com/survivejs/webpack-merge
    configureWebpack:{
        // &#x65B0;&#x589E;&#x63D2;&#x4EF6;&#x7B49;
        plugins:[]
    },
    devServer:{ // &#x914D;&#x7F6E;&#x4EE3;&#x7406;
        proxy:{
            '/api':{
                target:'http://a.zf.cn:3000',
                changeOrigin:true
            }
        }
    },
    // &#x7B2C;&#x4E09;&#x65B9;&#x63D2;&#x4EF6;&#x914D;&#x7F6E;
    pluginOptions: {
        'style-resources-loader': {
            preProcessor: 'less',
            patterns: [
                // &#x63D2;&#x5165;&#x5168;&#x5C40;&#x6837;&#x5F0F;
                path.resolve(__dirname,'src/assets/common.less'),
            ],
        }
    }
}
```

defer & async / preload & prefetch

- defer 和 async 在网络读取的过程中都是异步解析
- defer是有顺序依赖的，async只要脚本加载完后就会执行
- preload 可以对当前页面所需的脚本、样式等资源进行预加载
- prefetch 加载的资源一般不是用于当前页面的，是未来很可能用到的这样一些资源

基于vue-cli编写组件

小球的滚动组件

- 更改小球的颜色

```
<template>

</template>
<script>
export default {
  name:'ScrollBall',
  props:{
      color:{
          type:String,
          default:'red'
      }
  }
}
</script>
<style lang="less">
  .ball{
      width:80px;
      height: 80px;
      border-radius: 50%;
      text-align:center;
      line-height:80px;
  }
</style>
```

- 球的滚动requestAnimationFrame

```
<scrollball color="red" :target="500" v-model="pos1"></scrollball>
<scrollball color="blue" :target="300" v-model="pos2"></scrollball>

<script>
export default {
    props:{
        value:{
            type:Number,
            default:0
        },
        target:{
            type:Number,
            required:true
        }
    },
    mounted(){
        let ele = document.getElementById(`ball${this._uid}`);
        let timer;
        let fn = ()=>{
            let left = this.value + 2;
            if(left > this.target){
                return cancelAnimationFrame(timer);
            }
            ele.style.transform = `translate(${left}px)`;
            this.$emit('input',left);
            timer = requestAnimationFrame(fn)
        }
        timer = requestAnimationFrame(fn)
    }
}
</script>
```

- 增加球的内容

```
<scrollball color="red" :target="500" v-model="pos">&#x7403;1</scrollball>
```

- 让小球停止运动 增加小球的停止方法,通过父亲用$refs获取子组件方法

```
<scrollball color="red" :target="500" v-model="pos1" ref="ball">&#x7403;1</scrollball>
<button @click="stop">stop</button>
stop(){
  this.$refs.ball.stopMove()
}
// &#x7EC4;&#x4EF6;&#x4E2D;&#x505C;&#x6B62;&#x5C0F;&#x7403;&#x8FD0;&#x52A8;
methods:{
  stopMove(){
      cancelAnimationFrame(this.timer);
  },
  move(){
      let ele = document.getElementById(`ball${this._uid}`);
      let left = this.value + 2;
      if(left > this.target){
          return this.stopMove();
      }
      ele.style.transform = `translate(${left}px)`;
      this.$emit('input',left);
      this.timer = requestAnimationFrame(this.move)
  }
},
mounted(){
  this.timer = requestAnimationFrame(this.move)
}
```

- 通知小球运动结束

```
if(left > this.target){
  this.$emit('end');
  return this.stopMove();
}
```