《Vue最佳实践》

svg-sprite-loader

1.Webpack.module.rules配置

{

                test: /\.svg$/,

                loader: 'svg-sprite-loader',

                include: [resolve('src/icons')],

                options: {

                    symbolId: 'icon-[name]'

                }

            },

            {

                test: /\.(png|jpe?g|gif|svg)(\?.\*)?$/,

                loader: 'url-loader',

                exclude: [resolve('src/icons')],

                options: {

                    limit: 10000,

                    name: utils.assetsPath('img/[name].[hash:7].[ext]')

                }

            },

2.svgicon-component

<template>

  <svg :class="svgClass" aria-hidden="true">

    <use :xlink:href="iconName"></use>

  </svg>

</template>

<script>

export default {

  name: 'svg-icon',

  props: {

    iconClass: {

      type: String,

      required: true

    },

    className: {

      type: String

    }

  },

  computed: {

    iconName() {

      return `#icon-${this.iconClass}`

    },

    svgClass() {

      if (this.className) {

        return 'svg-icon ' + this.className

      } else {

        return 'svg-icon'

      }

    }

  }

}

</script>

<style scoped>

.svg-icon {

  width: 1em;

  height: 1em;

  vertical-align: -0.15em;

  fill: currentColor;

  overflow: hidden;

}

</style>

3.注册 icons/index.js 动态引入svg

作用在于虽然全局组册了svg但是在当前组件调用某个svg还是需要import这个文件，我们可以通过webpack的context.require()来解决这个问题。

require.context('./svg', false, /.svg$/),第一个参数是引入文件所在的文件夹，第二个是否引入子文件夹，第三个是匹配规则，通过这个方法就可以动态的引入svg文件了。

.keys()返回svg数组

.map()再获取和上下文匹配的svg文件

import Vue from 'vue'

import SvgIcon from '@/components/SvgIcon'*// svg组件*

*// register globally*

Vue.component('svg-icon', SvgIcon)

const requireAll = requireContext => requireContext.keys().map(requireContext)

const req = require.context('./svg', false, /\.svg$/)

requireAll(req)

4.main.js

import '@/icons'; *// icon*

