

# Vue3 + Vite + TS项目引入iconfont图标 (S...

## 前言

每一个项目都避免不了使用各种各样的图标，如果我们使用了UI组件库，比如说ELement等，那么组件库有一些封装好的图标供我们使用。但是项目是多变的和复杂的，组件库提供的图标很多时候不能满足需求，这个时候就需要我们自己引入想要的图标了。

今天介绍的便是如何将iconfont阿里图标库的图标引入到我们的Vue3项目中来！

## 1.常用图标格式简介

我们使用图标的方式有很多种，比如说直接使用png、jpg等图片格式，这是最原始的方式，所以我们这里不过多阐述，它们的优缺点相信大家也知道。

这里我们主要对比阿里图标库提供3种图标引用方式：**unicode**、**font-class**、**symbol**。

### 1.1 unicode格式

这种方式比较原始，主要是将图标以字体的形式渲染到网页上，既然是以字体的格式，那么它的优缺点也是显而易见的。

特点：

- 兼容性最好，支持ie6+，及所有现代浏览器。
- 支持按字体的方式去动态调整图标大小，颜色等等。
- 但是因为是字体，所以不支持多色。只能使用平台里单色的图标，就算项目里有多色图标也会自动去色。

所以目前来说，我们并不建议以Unicode的方式引入iconfont图标。

使用方式示例代码：

```
1 <i class="iconfont">&#x33;</i>
```

### 1.2 font-class方式

font-class方式本质上还是使用的字体的形式，所以它的优缺点其实和unicode方式差不多，但是却运用很广泛，我们可以来看看它的特点，就知道为什么了。

特点：

- 兼容性良好，支持ie8+，及所有现代浏览器。
- 相比于unicode语意明确，书写更直观。可以很容易分辨这个icon是什么。
- 因为使用class来定义图标，所以当要替换图标时，只需要修改class里面的unicode引用。
- 不过因为本质上还是使用的字体，所以多色图标还是不支持的。

使用方式示例代码：

```
1 <i class="iconfont icon-xxx"></i>
```

可以看到它最大的特点就是语义明确，所以这是它很流行的重要原因之一。

### 1.3 symbol 方式

这种方式相较于而言比较新，它主要利用了svg这种格式，在以前，由于它的浏览器兼容性不太好，所以没有大规模的流行。不过就目前来看，兼容性基本上不是问题了，毕竟IE都淘汰了。所以它是目前比较推崇的图标引用格式，至于为什么，我们一起来看看它的特点。

特点：

- 支持多色图标了，不再受单色限制。
- 通过一些技巧，支持像字体那样，通过 `font-size`，`color` 来调整样式。
- 兼容性较差，支持 ie9+,及现代浏览器。
- 浏览器渲染svg的性能一般，还不如png。

使用示例代码：

```
1 <svg class="icon" aria-hidden="true">
2   <use xlink:href="#icon-xxx"></use>
3 </svg>
```

可以看到它的最大特点就是支持多色图标了，而且还支持向字体那样调整样式了，虽然它的渲染性能不及png，但是它的其它好处可比png多多了，所以我们通常建议项目中使用symbol的方式引入图标库。

这也是本篇文章将要讲解的方式。

## 2.环境准备

为了方便演示以及让大家更容易理解，我直接使用 vite 搭建一个最基本的 Vue3 项目。

创建命令：

```
1 npm create vite@latest my-vite-app --template vue-ts
```

删除 App.vue 中一些不需要的东西，然后运行项目：



## 3.引入iconfont

### 3.1 封装svg-icon组件

我们在iconfont官网上可以看到给出了symbol引用示例，代码如下：

```
1 <svg class="icon" aria-hidden="true">
2   <use xlink:href="#icon-xxx"></use>
3 </svg>
```

上段代码就是图标的具体使用方式，如果我们每次都按照上面的方式使用，那么无疑是很麻烦的，我们不妨将上面的代码封装为一个组件。在需要用到图标的地方直接引入我们的组件即可了。

在components目录下新建SvgIcon目录，然后新建SvgIcon.vue文件。

代码如下：

```
1 // src/components/SvgIcon/SvgIcon.vue
2
3 <template>
4   <svg :class="svgClass" aria-hidden="true">
5     <use :xlink:href="iconClassName" :fill="color" />
6   </svg>
7 </template>
8
9 <script setup lang="ts">
10  import { computed } from 'vue';
11  const props = defineProps({
12    iconName: {
```

```

13     type: String,
14     required: true
15   },
16   className: {
17     type: String,
18     default: ''
19   },
20   color: {
21     type: String,
22     default: '#409eff'
23   }
24 });
25 // 图标在iconfont中的名字
26 const iconClassName = computed(()=>{
27   return `#${props.iconName}`;
28 })
29 // 给图标添加上类名
30 const svgClass = computed(() => {
31   if (props.className) {
32     return `svg-icon ${props.className}`;
33   }
34   return 'svg-icon';
35 });
36 </script>
37
38 <style scoped>
39 .svg-icon {
40   width: 1em;
41   height: 1em;
42   position: relative;
43   fill: currentColor;
44   vertical-align: -2px;
45 }
46 </style>

```

组件封装好后我们还需要全局注册一下，不然每次引用图标的时候还得单独引入一次该组件。所以我们就在main.ts里面全局注册一下。

代码如下：

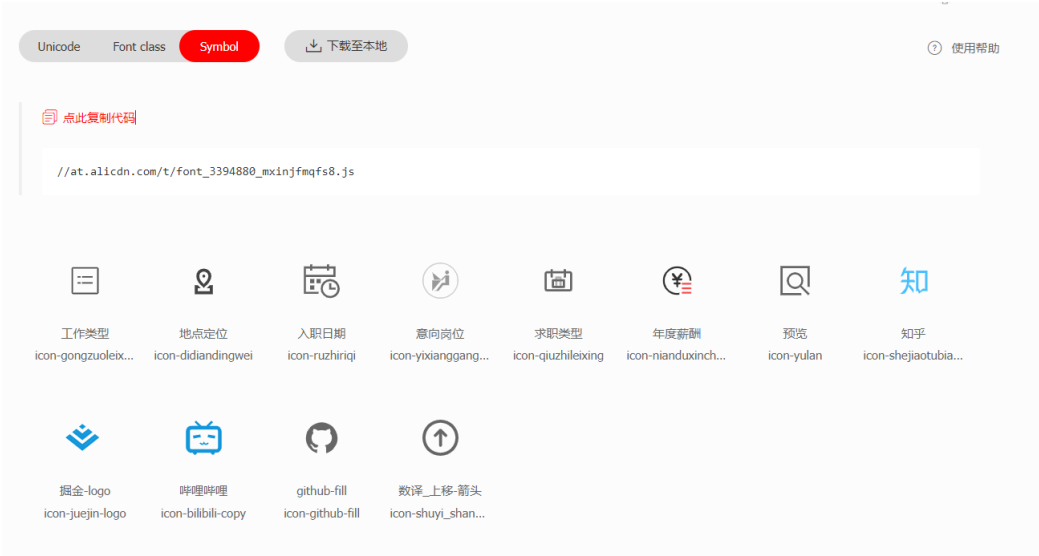
```

1 // main.ts
2 import { createApp } from 'vue'
3 import App from './App.vue'
4 import SvgIcon from './components/SvgIcon/SvgIcon.vue'
5
6 const app = createApp(App)
7 app.component('SvgIcon', SvgIcon);
8 app.mount('#app')

```

### 3.2 引入iconfont

接下来我们就需要去iconfont官网了，新建一个iconfont资源库，存放自己的iconfont。如下图所示：



我们选中symbol模式，这里我介绍三种引入方式。

**第一种：**

直接在线引入官网提供的在线js地址，我们直接以script标签的形式引入即可。这种方式最为简单，但是也有不好的一点，需要用户有网络环境，而且得保证iconfont网站没有崩掉。

**第二种：**

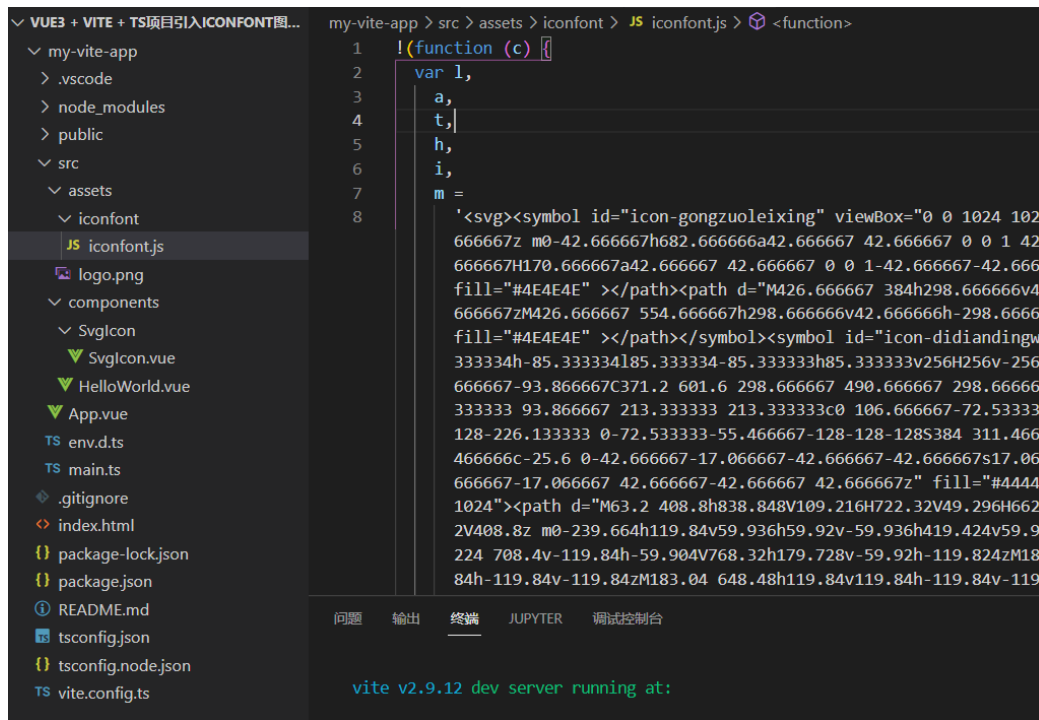
直接下载至本地，我们从官网直接将代码下载下来，然后放到我们项目中引用，也是可以的。不过这种方式稍显麻烦，每次更新图标库之后都得重新下载一遍。

**第三种：**

这也是我比较喜欢的方式，也就是将在线地址中的代码直接复制粘贴到我们项目中来，这种方式最为简单，每次更新图标库之后只需要重新复制一下代码即可。这里我们也将采用这种方式。

**具体使用：**

在项目assets目录下新建iconfont目录，在该目录下新建iconfont.js文件，然后将iconfont在线地址中提供的代码全部复制过来。



然后在main.ts中引入iconfont.js。

代码如下：

```
1 import { createApp } from 'vue'
2 import App from './App.vue'
3 import SvgIcon from './components/SvgIcon/SvgIcon.vue'
4 import './assets/iconfont/iconfont.js';
5
6 const app = createApp(App)
7
8 app.component('SvgIcon', SvgIcon);
9 app.mount('#app')
```

## 4.具体使用

引入iconfont得过程非常简单，主要分为了以下两步：

- 封装svg-icon组件
- 引入iconfont生成的js代码

接下来我们就实际使用试试，我们就直接子啊App.vue中引入几个图标试试。

代码如下：

```
1 <template>
2   
3   <svg-icon iconName="icon-gongzuoleixing"></svg-icon>
4   <svg-icon iconName="icon-yulan" className="yulan"></svg-icon>
5 </template>
```

输出结果：



```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <svg aria-hidden="true" style="position: absolute; width: 0px; height:
    x; overflow: hidden;">...</svg>
    <div id="app" data-v-app> (flex)
      
    ...
    <svg class="svg-icon" aria-hidden="true" data-v-9e6ba36a>...</svg> == $
    <svg class="svg-icon yulan" aria-hidden="true" data-v-9e6ba36a>...</svg>
    </div>
    <script type="module" src="/src/main.ts?t=1655103736303"></script>
    <div class="xl-chrome-ext-bar" id="xl_chrome_ext_{40B361DE-01F7-4376-B4
    -639E489D19ED}" style="display: none;">...</div>
    <div id="mbGetImgModule" class="mb-getmaterial-box" style="display: nor
    ">...</div>
  </body>
</html>
```

可以看到我们的图标已经可以使用了，其中iconName属性值就是我们在iconfont网站复制的iconfont的名称代码。如果想要该颜色或大小，可以自行传入一个类或者color属性。

除此之外，如果你有自己下载好的svg文件，那么也是可以通过上面方式引入的，只需要将iconName改为你自己本地的svg名称即可。

## 总结

图标是一个项目中必不可少的东西，引入图标经常是在我们项目搭建的时候就该完成的，不管你是使用哪种引入图标的方式，都应该遵循一个原则：方便、易理解。