

Vue.js中引入外部JS：简单方法，灵活选择

作者：微信公众号：【架构师老卢】

12-21 14:40

12



概述：在Vue.js组件中添加外部JS脚本有多种方法，包括使用`mounted`生命周期钩子、`vue-plugin-load-script`插件以及`import`语句。每种方法都提供了灵活性和便利性，可根据项目需求选择适合的方式。

在Vue.js组件中添加外部JS脚本可以通过以下几种方法实现。下面将详细介绍每种方法的步骤以及相关的实例源代码。

方法一：使用mounted生命周期钩子

在Vue.js组件中，可以使用mounted生命周期钩子来在组件被挂载后执行一些操作，包括加载外部JS脚本。

```
1 <template>
2   <div>
3     <!-- 组件内容 -->
4   </div>
5 </template>
6
7 <script>
8 export default {
9   mounted() {
10    this.loadExternalScript();
11  },
12  methods: {
13    loadExternalScript() {
14      // 创建script标签
15      const script = document.createElement('script');
16      script.src = 'external-script.js'; // 外部JS脚本的路径
17
18      // 将script标签添加到body中
19      document.body.appendChild(script);
20    }
21  }
22 };
23 </script>
```

方法二：使用vue-plugin-load-script插件

vue-plugin-load-script是一个专门用于在Vue.js中加载外部脚本的插件，使用起来更加方便。

首先，安装插件：

```
1 npm install vue-plugin-load-script
```

然后在组件中使用：

```
1 <template>
2   <div>
3     <!-- 组件内容 -->
4   </div>
5 </template>
6
7 <script>
8 import Vue from 'vue';
9 import LoadScript from 'vue-plugin-load-script';
10
11 Vue.use(LoadScript);
12
13 export default {
14   mounted() {
15     this.$loadScript('external-script.js')
16       .then(() => {
17         console.log('外部脚本加载成功！');
18       })
19       .catch(() => {
20         console.error('外部脚本加载失败！');
21       });
22   }
23 };
24 </script>
```

方法三：使用import语句

如果外部JS脚本是ES6模块，也可以使用import语句直接在组件中导入。

```
1 <template>
2   <div>
3     <!-- 组件内容 -->
4   </div>
5 </template>
6
7 <script>
8 import ExternalScript from 'external-script'; // 外部JS脚本的路径
9
10 export default {
11   mounted() {
12     // 使用导入的外部脚本
13     ExternalScript.someFunction();
14   }
15 }
```

```
15 | };  
16 | </script>
```

选择其中一种方法，根据项目的实际情况和需求来决定使用哪种方式。