## Monaco Editor 性能优化方案

**一、总体思想：减少体积 + 按需加载 + 预加载**

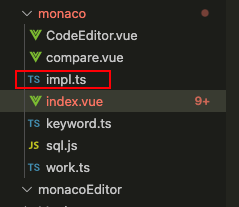
1. 减少体积：根据需要，按需引入语言包；
2. 预加载：使用魔法注释实现预加载；
3. 按需加载：使用import()实现异步按需加载；

import(/\* webpackChunkName: "monaco-editor", webpackPrefetch: true \*/ 'monaco-editor')

1. **需要协调处理的问题**

需要前端同时协调处理的是：第3点按需加载。

项目现在存在的问题是：多个文件中有引入 Monaco 的代码，比如 impl.ts、index.vue、worker.ts 中都有 import \* as monaco from 'monaco-editor/esm/vs/editor/editor.api'; 的语句。



需要改造的内容

**导入：**只在一个js/ts 有 import \* as monaco from 'monaco-editor/esm/vs/editor/editor.api'; 这样的语句。然后在引入的 monaco 上添加方法，最后将 Monaco 和这些方法导出。

**导出：**在需要使用Monaco和其方法的地方使用异步的方式导入，同时做好边界处理。