

[🏠](#) / [开发指南](#) / [前端手册 Vue 3.x](#)[👤 芋道源码](#) [📅 2023-01-01](#)

🔗 通用方法

本小节，分享前端项目的常用方法。

1. 缓存配置

友情提示：

该小节，基于 [《vue element plus admin —— 项目配置「缓存配置」》](#) 的内容修改。

1.1 说明

在项目中，你可以看到很多地方都使用了 `wsCache.set` 或者 `wsCache.get`，这是基于 [web-storage-cache](#) 进行封装，采用 `hook` 的形式。

该插件对HTML5 `localStorage` 和 `sessionStorage` 进行了扩展，添加了超时时间，序列化方法。可以直接存储 `json` 对象，同时可以非常简单的进行超时时间的设置。

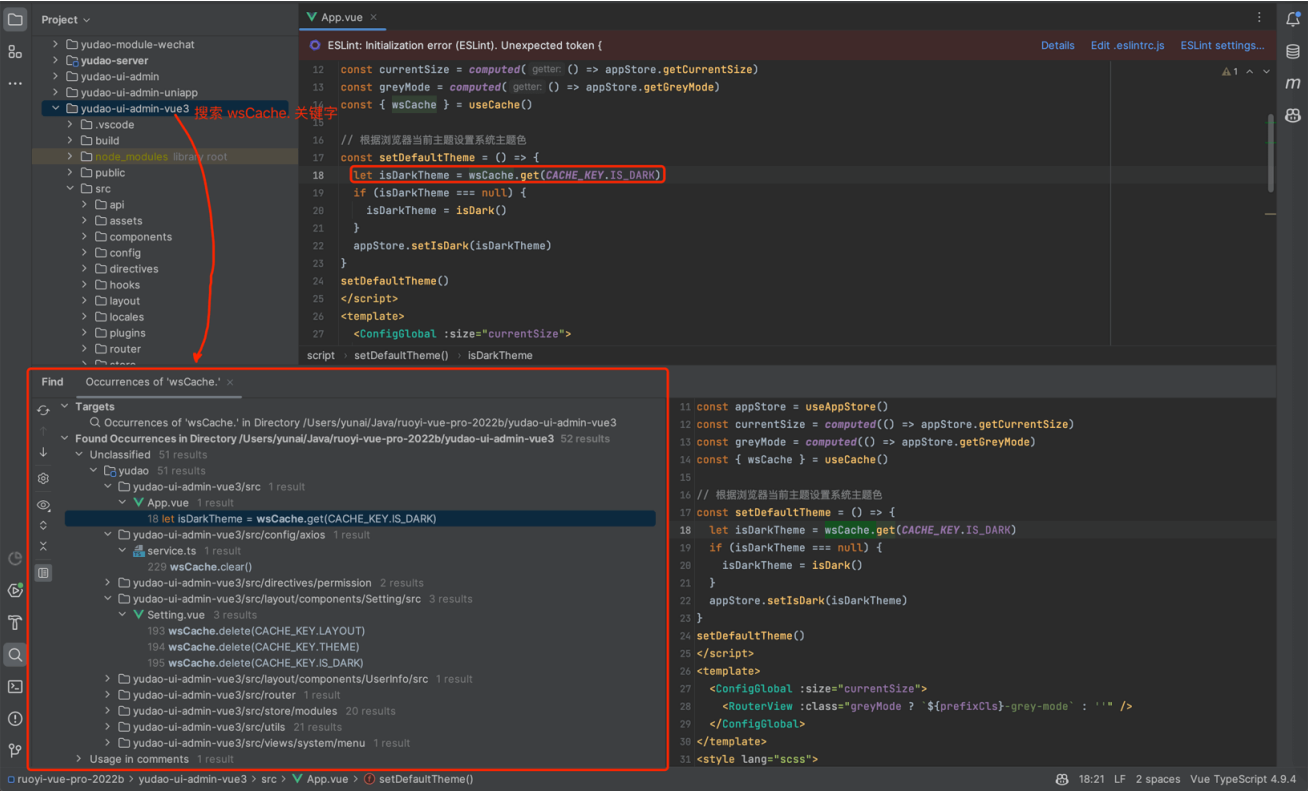
本项目默认是采用 `sessionStorage` 的存储方式，如果更改，可以直接在 [useCache.ts](#) 中把 `type: CacheType = 'sessionStorage'` 改为 `type: CacheType = 'localStorage'`，这样项目中的所有用到的地方，都会变成该方式进行数据存储。

如果只想单个更改，可以传入存储类型 `const { wsCache } = useCache('localStorage')`，既可只适用当前存储对象。

注意：

更改完默认存储方式后，需要清除浏览器缓存并重新登录，以免造成不可描述的问题。

1.2 示例

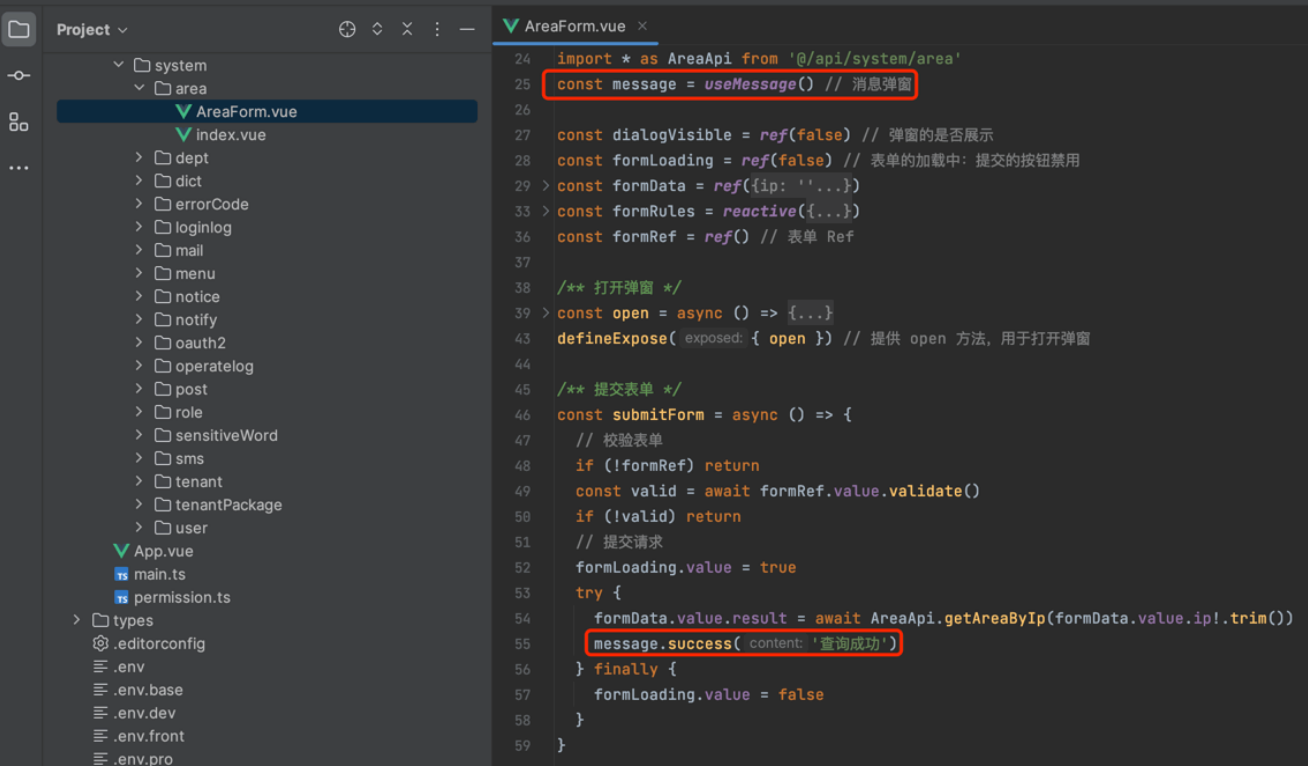


2. message 对象

2.1 说明

message 对象，由 [src/hooks/web/useMessage.ts](#) 实现，基于 EIMessage、EIMessageBox、EINotification 封装，用于做消息提示、通知提示、对话框提醒、二次确认等。

2.2 示例



3. download 对象

3.1 说明

`$download` 对象, 由 `util/download.ts` 实现, 用于 Excel、Word、Zip、HTML 等类型的文件下载。

3.2 示例

Project

views

bpm

Error

Home

infra

Login

mall

mp

pay

Profile

Redirect

report

system

area

AreaForm.vue

index.vue

dept

DeptForm.vue

index.vue

dict

errorCode

loginlog

index.vue

LoginLogDetail.vue

mail

menu

notice

notify

oauth2

operatelog

post

role

sensitiveWord

sms

tenant

tenantPackage

user

App.vue

main.ts

permission.ts

types

editorconfig

env

env.base

index.vue

```
187 <script setup lang="ts" name="SystemLoginLog">
188 import { DICT_TYPE } from '@/utils/dict'
189 import { dateFormatter } from '@/utils/formatTime'
190 import download from '@/utils/download'
191 import * as LoginLogApi from '@api/system/LoginLog'
192 import LoginLogDetail from './LoginLogDetail.vue'
193 const message = useMessage() // 消息弹窗
194
195 const loading = ref(true) // 列表的加载中
196 const total = ref(0) // 列表的总页数
197 const list = ref([]) // 列表的数据
198 > const queryParams = reactive({username: undefined...})
199 const queryFormRef = ref() // 搜索的表单
200 const exportLoading = ref(false) // 导出的加载中
201
202 /** 查询列表 */
203 > const getList = async () => {...}
204
205 /** 搜索按钮操作 */
206 > const handleQuery = () => {...}
207
208 /** 重置按钮操作 */
209 > const resetQuery = () => {...}
210
211 /** 详情操作 */
212 const detailRef = ref()
213 > const openDetail = (data: LoginLogApi.LoginLogVO) => {...}
214
215 /** 导出按钮操作 */
216 > const handleExport = async () => {
217   try {
218     // 导出的二次确认
219     await message.exportConfirm()
220     // 发起导出
221     exportLoading.value = true
222     const data = await LoginLogApi.exportLoginLog(queryParams)
223     download.excel(data, {fileName: '登录日志.xls'})
224   } catch {
225
226   } finally {
227     exportLoading.value = false
228   }
229 }
230 }
```

