

Vue 项目国际化高效实践指南

一、背景与目标

当前项目存在多个端：PC 端（Vue 3）、移动 App 与企业微信端（Vue 2），均已引入 `vue-i18n`，但部分页面中文写死，维护成本高，翻译流程零散、重复、低效。

本指南旨在为团队建立一套高效、统一的国际化自动化流程：

- 自动提取硬编码中文
- 自动替换为 `$t()` 并生成 key
- 自动生成翻译文件及导出 Excel
- 跨端协作统一标准

二、技术选型

模块	工具/库
国际化框架	<code>vue-i18n</code> (Vue 2 & Vue 3)
自动提取中文	@wansho/i18n-extract
自动替换中文	i18n Ally 插件
翻译协作导出	Excel (由 JSON 转换)

三、整体流程概览

✓ 步骤流程：

- 使用 i18n Ally 自动识别中文 → 替换为 `$t('key')`
- 使用 @wansho/i18n-extract 扫描源码生成 JSON 语言包
- 将语言包导出为 Excel 供翻译
- 翻译后同步更新语言包，统一交付

四、配置与使用详解

4.1 各端引入 vue-i18n

✓ Vue 3 (PC端)

```
import { createI18n } from 'vue-i18n';
import zh from './locales/zh.json';
import en from './locales/en.json';

const i18n = createI18n({
  legacy: false,
  locale: 'zh',
  messages: { zh, en },
});
```

```
app.use(i18n);
```

✅ Vue 2 (移动端/企业微信)

```
import VueI18n from 'vue-i18n';
import Vue from 'vue';
import zh from './locales/zh.json';
import en from './locales/en.json';

Vue.use(VueI18n);

const i18n = new VueI18n({
  locale: 'zh',
  messages: { zh, en },
});

new Vue({ i18n }).$mount('#app');
```

4.2 使用 `$t()` 替换中文字段

```
<!-- 替换前 -->
<p>欢迎使用</p>
<!-- 替换后 -->
<p>{{ $t('home.welcome') }}</p>
```

五、工具使用详解

5.1 使用 i18n Ally 快速替换中文

- 安装插件: [i18n Ally](#)
- 在设置中配置语言包路径 (如 `src/locales/zh.json`)
- 打开 `.vue/.ts` 文件: 插件自动高亮中文 → 右键“Extract to \$t”
- 自动生成 key 并加入语言包 JSON 文件

⚠️ 支持 Vue 2/3 单文件组件和 TS 文件, 适合批量处理历史遗留代码

5.2 使用 @wansho/i18n-extract 自动生成翻译包

安装依赖

```
npx @wansho/i18n-extract init
```

扫描代码目录

```
npx @wansho/i18n-extract --dir src
```

- 自动扫描所有中文文本 → 替换为 `$t('key')`
- 生成 `zh.json`, `en.json`
- 可导出为 Excel 文件进行翻译协作

六、翻译协作流程

1. 每位开发者只提交中英文 JSON 文件，不提交写死中文
2. 翻译由产品或运营导出 Excel 完成后统一回填
3. 合并翻译后 JSON，保证多端统一
4. 定期对项目运行 `extract`，补全遗漏中文

Excel 表格结构

key	中文	英文
home.welcome	欢迎使用	Welcome
login.submit	提交	Submit

七、命名规范与建议

- 模块名.字段名 命名方式（如 `login.username`，`common.submit`）
- 公共字段统一放入 `common` 模块
- 禁止混用 `$t()` 与中文硬编码
- 建议结合 ESLint 检查器，强制禁用中文文本

八、常见问题

◆ `$t` 是什么？

`$t()` 是 `vue-i18n` 提供的翻译函数，作用是读取配置的翻译文件（JSON），输出对应语言内容。

```
$t('home.welcome') // -> “欢迎使用” 或 “welcome”
```

◆ 项目中没用 `$t` 怎么办？

- 使用 `i18n Ally` 插件 → 自动替换为 `$t()`
- 使用 `i18n-extract` → 自动生成语言包

九、参考资料

- [vue-i18n 官方文档](#)
- [i18n Ally 插件](#)
- [i18n-extract GitHub](#)
- [掘金 Vue 国际化实战](#)

十、结语

国际化是中小型项目跨国落地的基础能力，本指南致力于让你从“重复劳动”中解放出来，高效完成 `i18n` 改造与维护。未来还可引入 CI 自动校验缺失字段、云端翻译平台协作，打造真正工程化的国际化体系。