

国际化

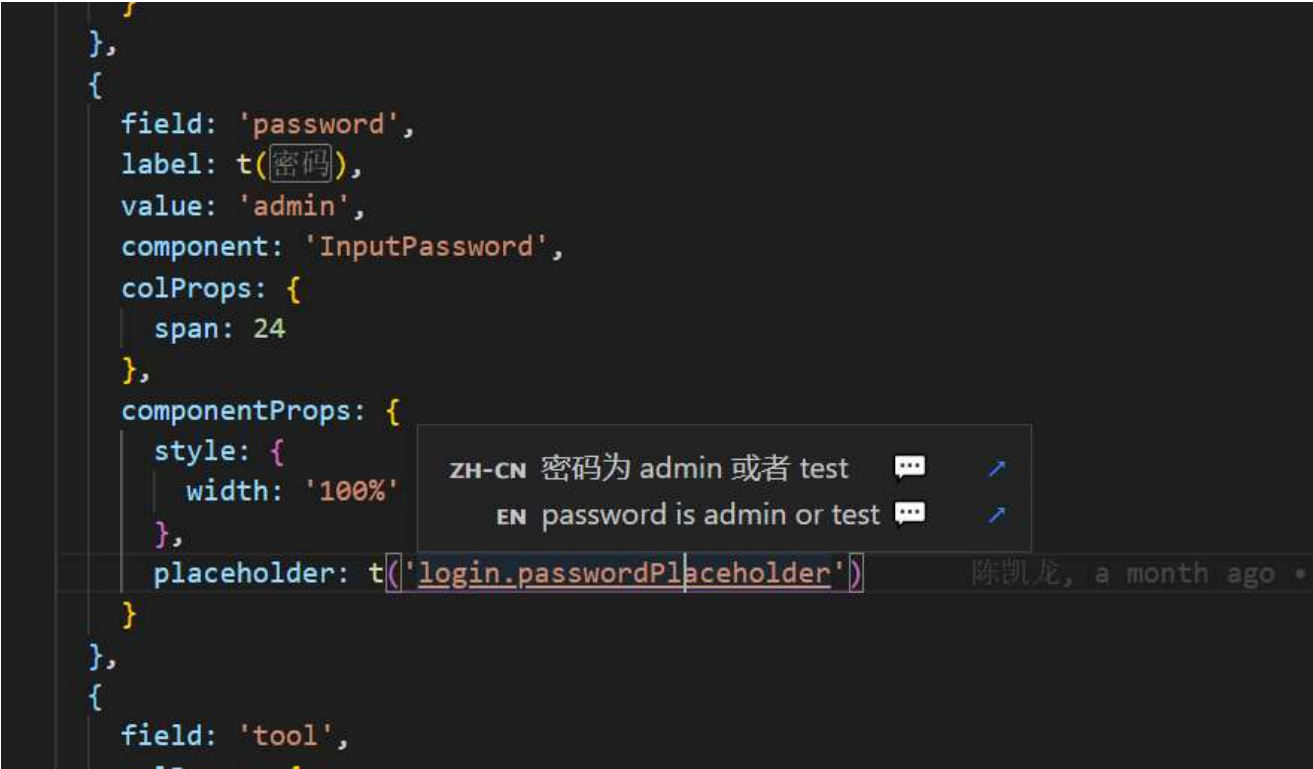
友情提示:

该章节，基于 [《vue element plus admin —— 国际化》](#) 的内容修改。

如果你使用的 vscode 开发工具，则推荐安装 [l18n-ally](#) 这个插件

1. l18n-ally 插件

安装了该插件后，你的代码内可以实时看到对应的语言内容



VS Code 代码编辑器截图，展示了 l18n-ally 插件的实时翻译功能。代码片段如下：

```
},
{
  field: 'password',
  label: t('密码'),
  value: 'admin',
  component: 'InputPassword',
  colProps: {
    span: 24
  },
  componentProps: {
    style: {
      width: '100%'
    },
    placeholder: t('login.passwordPlaceholder')
  }
},
{
  field: 'tool',
```

在 `t('login.passwordPlaceholder')` 处，弹出了一个语言选择框，显示了两种语言版本：

- ZH-CN 密码为 admin 或者 test
- EN password is admin or test

右侧显示了用户信息：陈凯龙, a month ago。

2. 配置默认语言

在 [src/store/modules/locale.ts](#) 内配置 `currentLocale` 为其他语言。

► [查看代码](#)

3. 语言文件

在 [src/locales](#) 可以配置具体的语言。

目前项目中的语言都是没有拆分的，全部放一起，后续会考虑拆分出来，比较好维护。

4. 语言导入逻辑说明

在 `src/plugins/vuel18n/index.ts` 内可以看到

```
const defaultLocal = await import(`../../locales/${locale.lang}.ts`)
```

这会导入 `src/locales` 文件语言包。

5. 使用

引入项目自带的 `useI18n`

注意不要引入 vue-i18n 的 useI18n

```
import { useI18n } from '@/hooks/web/useI18n'

const { t } = useI18n()

const title = t('common.menu')
```

6. 切换语言

切换语言需要使用 `src/hooks/web/useLocale.ts`

```
import { useLocale } from '@/hooks/web/useLocale'
const { changeLocale } = useLocale()

changeLocale('en')
```

7. 新增新语言

7.1 语言文件

在 `src/locales` 增加对应语言的文件即可

7.2 新增语言

目前项目自带的语言只有 `zh_CN` 和 `en` 两种

如果需要新增，按以下操作即可

1. 在 `src/locales` 下语言文件

2. 在 [types/global.d.ts](#) 给 `LocaleType` 添加对应的类型
3. 在 [src/store/modules/locale.ts](#) `localeMap` 中添加对应语言

8. 远程读取语言数据

目前项目会在 `src/main.ts` 内等待 `setupI18n` 这个函数执行完之后才会渲染界面，所以只需在 `setupI18n` 内的 `createI18nOptions` 发送 ajax 请求，将对应的数据设置到 `i18n` 实例上即可。

```
const createI18nOptions = async (): Promise<I18nOptions> => {
  const localeStore = useLocaleStoreWithOut()
  const locale = localeStore.getCurrentLocale
  const localeMap = localeStore.getLocaleMap
  // 这里改为远程请求即可。
  const defaultLocal = await import(`../../locales/${locale.lang}.ts`)
  const message = defaultLocal.default ?? {}

  setHtmlPageLang(locale.lang)

  localeStore.setCurrentLocale({
    lang: locale.lang
    // elLocale: elLocal
  })

  return {
    legacy: false,
    locale: locale.lang,
    fallbackLocale: locale.lang,
    messages: {
      [locale.lang]: message
    },
    availableLocales: localeMap.map((v) => v.lang),
    sync: true,
    silentTranslationWarn: true,
    missingWarn: false,
    silentFallbackWarn: true
  }
}
```

8.1 useLocale

代码: [src/hooks/web/useLocale.ts](#)

当手动切换语言的时候会触发 `useLocale` 函数, `useLocale` 也是异步函数, 只需等待接口返回响应的数据后, 再进行设置即可

```
export const useLocale = () => {
  // Switching the language will change the locale of useI18n
  // And submit to configuration modification
  const changeLocale = async (locale: LocaleType) => {
    const globalI18n = i18n.global

    // 改为远程获取
    const langModule = await import(`../../locales/${locale}.ts`)

    globalI18n.setLocaleMessage(locale, langModule.default)

    setI18nLanguage(locale)
  }

  return {
    changeLocale
  }
}
```

[← CRUD 组件](#)[IDE 调试→](#)

Theme by **Vdoing** | Copyright © 2019-2024 芋道源码 | MIT License