

# vue4.0模板文档(包含多语言、px转rem、路...

## 一、多语言 (Vue 4.0, 示例为中英文)

### 1. 安装插件

`npm install vue-i18n --save-dev`

### 2. 在目录src下新建目录locales, locales目录下新建文件en.js、zh.js

en.js文件内容:

```
1 module.exports = {  
2   message: 'test!'  
3 };
```

zh.js文件内容:

```
1 module.exports = {  
2   message: '测试!'  
3 };
```

### 3. 在文件constants/index.js中增加

#### 1. 文件constants/index.js不存在时:

在目录src下新建目录constants, 目录constants下新建index.js, index.js文件内容如下:

```
1 module.exports = {  
2   // 默认语言  
3   DEFAULT_LOCALE: 'zh'  
4 };
```

在src/main.js中增加:

```
1 import constants from './constants';  
2 Vue.prototype.$const = constants;
```

#### 2. 文件constants/index.js存在时:

在constants/index.js中增加:

```
1 // 默认语言  
2 DEFAULT_LOCALE: 'zh'
```

### 4. 在目录src下新建文件i18n.js

文件内容：

```
1 import Vue from 'vue';
2 import VueI18n from 'vue-i18n';
3 import CONST from '@/constants';
4 Vue.use(VueI18n);
5 function loadLocaleMessages () {
6   const locales = require.context('./locales', true, /[A-Za-z0-9-
7   _,\s]+\./i);
8   const messages = {};
9   locales.keys().forEach(key => {
10     const matched = key.match(/([A-Za-z0-9-_]+\./i);
11     if (matched && matched.length > 1) {
12       const locale = matched[1];
13       messages[locale] = locales(key);
14     }
15   });
16   return messages;
17 };
18 export default new VueI18n({
19   locale: process.env.VUE_APP_I18N_LOCALE || CONST.DEFAULT_LOCALE,
20   fallbackLocale: process.env.VUE_APP_I18N_FALLBACK_LOCALE || CONST.DEFAULT_LOCALE,
21   messages: loadLocaleMessages()
22 });
```

## 5. 在src/main.js中增加：

```
1 import i18n from './i18n';
2 new Vue({
3   router,
4   store,
5   i18n,
6   render: h => h(App)
7 }).$mount('#app');
```

## 6. 修改src/router/index.js文件：

引入文件：

```
1 import i18n from '@/i18n';
2 import CONST from '@/constants';
```

修改routes：

```

1  const routes = [
2    {
3      path: '/:lang',
4      component: () => import('@App.vue'),
5      children: [
6        {
7          path: '',
8          redirect: 'home'
9        }, {
10         path: 'home',
11         name: 'Home',
12         component: () => import('@views/Home.vue')
13       }
14     ]
15   }
16 ];

```

增加：

```

1  router.beforeEach((to, from, next) => {
2    if (to.params.lang && i18n.availableLocales.includes(to.params.lang)) {
3      if (to.params.lang !== from.params.lang) {
4        i18n.locale = to.params.lang;
5      }
6      next();
7    } else {
8      next(`/${i18n.locale || CONST.DEFAULT_LOCALE}`);
9    }
10 });

```

## 7. 增加语言切换开关（示例在src/views/Home.vue文件中）

**<template></template>中：**

```

1  <div class="home-container">|
2    <template v-for="(lang, i) in $i18n.availableLocales">
3      <a :href="switchLang(lang)" :key="i">{{ lang }}</a> |
4    </template>
5  </div>

```

**<script></script>中：**

```

1  export default {
2    methods: {
3      switchLang(val) {

```

```

4      const me = this;
5      let routePathInfo = JSON.parse(JSON.stringify(me.$route.path));
6      let routeQueryInfo = window.location.search;
7      if (me.$route.params.lang && val !== me.$route.params.lang) {
8          let routePathList = routePathInfo.split('/');
9          if (routePathList.length > 1) {
10             routePathList[1] = val;
11         }
12         routePathInfo = routePathList.join('/');
13     }
14     return routePathInfo + routeQueryInfo;
15 },
16 },
17 };

```

## 8. 使用方法

```

1 <p>{{ $t('message') }}</p>

```

## 9. 效果



## 二、px转rem (Vue 4.0)

\*包含scss全局方法 (@function) 、全局变量 (\$test) 的配置和使用

\*包含js中使用scss变量的方法

1. 安装插件： (**sass-loader**必须是**较低版本**，否则会**报错**)

**npm install node-sass --save-dev**

**npm install sass-loader@7.1.0 --save-dev**

**npm install sass-resources-loader --save-dev**

2. 在目录src/assets下新建**目录styles**， styles目录下新建**文件**

**public.scss**

文件内容：

```

1 // 媒体查询 max-width
2 $mediaScreenMaxWidthOnlyNumber: 1200;
3 $mediaScreenMaxWidth: $mediaScreenMaxWidthOnlyNumber + px;
4 // 设计图宽度
5 $baseWidth: 750;
6 // $baseWidth/$baseSize 750/75 --- 10rem
7 $baseSize: 75;
8 // pc端baseSize
9 $pcBaseSize: 16;
10 @function px2rem($px) {
11     @return 1 / $baseSize * $px * 1rem;
12 }
13 :export {
14     mediaScreenMaxWidth: $mediaScreenMaxWidthOnlyNumber;
15     baseSize: $baseSize;
16     baseWidth: $baseWidth;
17     pcBaseSize: $pcBaseSize;
18 }

```

### 3. 在vue.config.js中增加：

```

1 css: {
2     loaderOptions: {
3         sass: {
4             //添加全局sass文件 此处路径替换为你的文件路径
5             data: `@import "~@/assets/styles/public.scss";`
6         }
7     }
8 }

```

**\*主要是以下配置**

```

1 sass: {
2     //添加全局sass文件 此处路径替换为你的文件路径
3     data: `@import "~@/assets/styles/public.scss";`
4 }

```

### 4. 在目录src下新建目录libs，libs目录下新建文件rem.js

文件内容：

```

1 import styles from '@/assets/styles/public.scss';
2 const baseSize = styles.baseSize;
3 const baseWidth = styles.baseWidth;

```

```

4  const pcBaseSize = styles.pcBaseSize;
5  const pxToRemWidth = styles.mediaScreenMaxWidth;
6  function setRem() {
7      if (document.documentElement.clientWidth > pxToRemWidth) {
8          document.documentElement.style.fontSize = pcBaseSize + 'px';
9      } else {
10         const scale = document.documentElement.clientWidth / baseWidth;
11         document.documentElement.style.fontSize = baseSize * Math.min(scale, 2) + 'px';
12     }
13 }
14 setRem();
15 window.onresize = function () {
16     setRem();
17 }

```

## 5. 在src/main.js中增加：

```

1  import './libs/rem.js'

```

## 6. 使用方法

1. 在scss文件中使用或vue文件中<style>标签增加lang="scss"后使用
2. 例：设置宽度

```

1  width: px2rem(750);

```

3. 例：媒体查询

```

1  @media screen and (max-width: $mediaScreenMaxWidth) {}

```

# 三、路由

## 1. 规则

### 一级路由：

示例：/news/index\_1

新闻页，page = 1

### 二级路由：

示例：/news/newest\_1\_20

新闻类别下的最新新闻页，page = 1，page\_size = 20