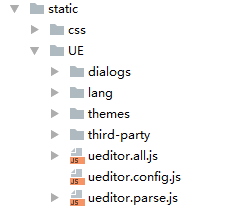
[vue2.0项目中使用Ueditor富文本编辑器示例](https://www.cnblogs.com/dmcl/p/7152711.html)

最近在vue项目中需要使用富文本编辑器，于是将Ueditor集成进来，作为公共组件。  
在线预览：https://suweiteng.github.io/vue2-management-platform/#/editor  
项目地址：https://github.com/suweiteng/vue2-management-platform 记得在右上角点个赞哦~

**1.放入静态资源并配置**

首先把官网下载的Ueditor资源，放入静态资源src/static中。  
  
修改ueditor.config.js中的window.UEDITOR\_HOME\_URL配置，如下图：  


**2.引入**

在main.js中引入

import '../static/UE/ueditor.config.js'

import '../static/UE/ueditor.all.min.js'

import '../static/UE/lang/zh-cn/zh-cn.js'

import '../static/UE/ueditor.parse.min.js'

**3.开发公共组件**

开发公共组件，可设置填充内容defaultMsg，配置信息config(宽度和高度等)，并提供获取内容的方法。

<template>

<div>

<script id="editor" type="text/plain"></script>

</div>

</template>

<script>

export default {

name: 'UE',

data () {

return {

editor: null

}

},

props: {

defaultMsg: {

type: String

},

config: {

type: Object

}

},

mounted() {

const \_this = this;

this.editor = UE.getEditor('editor', this.config); // 初始化UE

this.editor.addListener("ready", function () {

\_this.editor.setContent(\_this.defaultMsg); // 确保UE加载完成后，放入内容。

});

},

methods: {

getUEContent() { // 获取内容方法

return this.editor.getContent()

}

},

destroyed() {

this.editor.destroy();

}

}

</script>

**4.使用**

当我们需要使用富文本编辑器时，直接调用公共组件即可

<template>

<div class="components-container">

<div class="info">UE编辑器示例<br>需要使用编辑器时，调用UE公共组件即可。可设置填充内容defaultMsg，配置信息config(宽度和高度等)，可调用组件中获取内容的方法。</div>

<button @click="getUEContent()">获取内容</button>

<div class="editor-container">

<UE :defaultMsg=defaultMsg :config=config ref="ue"></UE>

</div>

</div>

</template>

<style>

.info{

border-radius: 10px;

line-height: 20px;

padding: 10px;

margin: 10px;

background-color: #ffffff;

}

</style>

<script>

import UE from '../../components/ue/ue.vue';

export default {

components: {UE},

data() {

return {

defaultMsg: '这里是UE测试',

config: {

initialFrameWidth: null,

initialFrameHeight: 350

}

}

},

methods: {

getUEContent() {

let content = this.$refs.ue.getUEContent();

this.$notify({

title: '获取成功，可在控制台查看！',

message: content,

type: 'success'

});

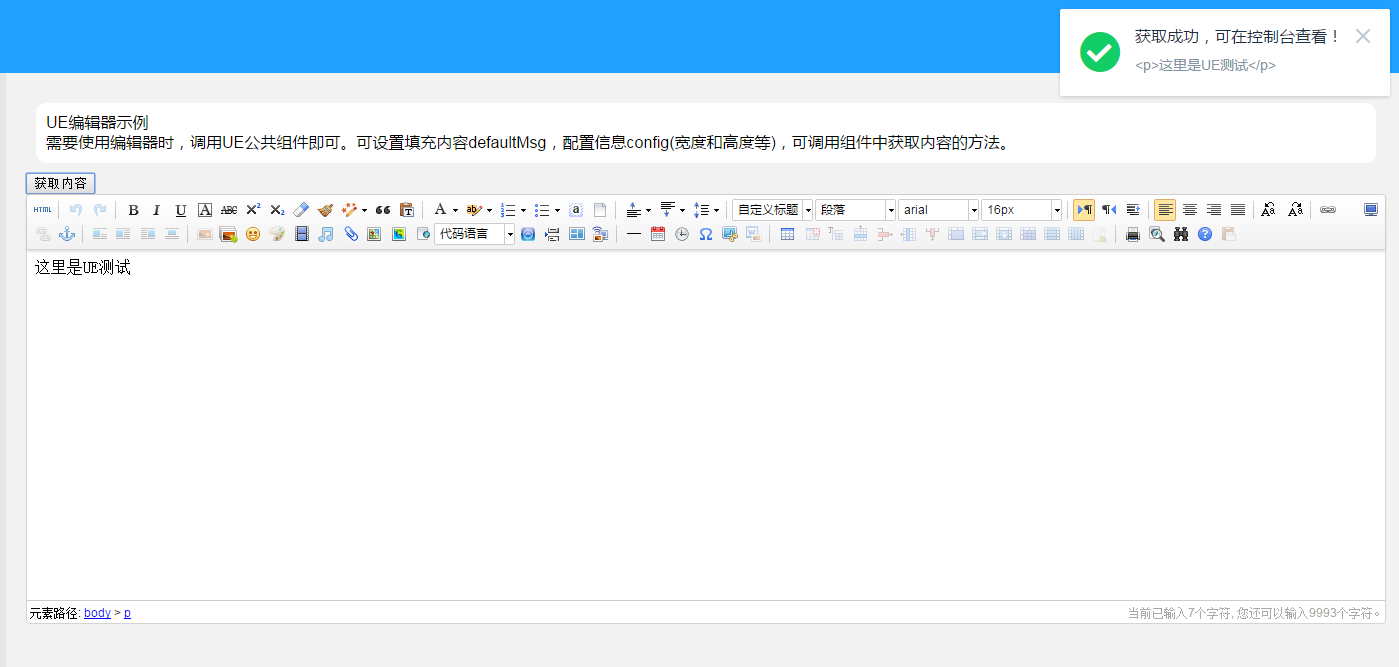
console.log(content)

}

}

};

</script>

效果如下：  


**5.报错**

**ESlint报错**

eslint报错的参考请评论4L 5L

**严格模式报错**

部分人使用时出现以下报错：  
Uncaught TypeError: 'caller', 'callee', and 'arguments' properties may not be accessed on strict mode functions or the arguments objects for calls to them...  
这个问题是因为项目中的使用的babel默认添加了use strict造成，可参考 https://segmentfault.com/q/1010000007415253  
我采用的是链接中答案的第三种方式：添加了babel-plugin-transform-remove-strict-mode，并在.babelrc里添加下列代码

{

"plugins": ["transform-remove-strict-mode"]

}

然后就没问题了。