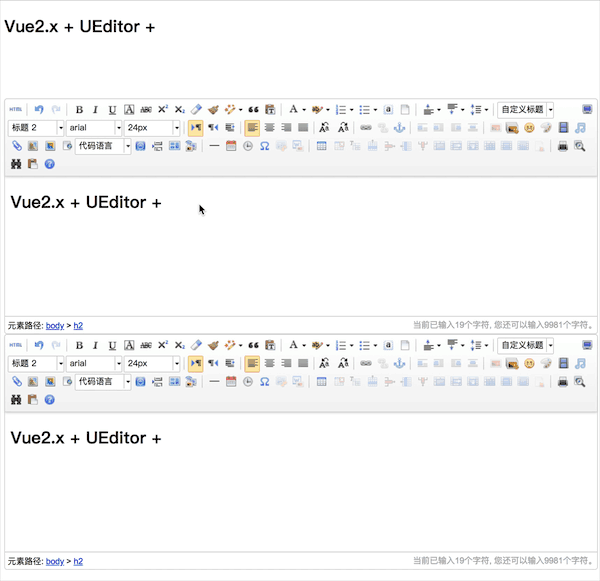
**ue-ueditor-wrap**

Vue + UEditor + v-model 双向绑定。之所以有这个 repo 的原因是：  
 1、UEditor 依然是国内使用频率极高的所见即所得编辑器而 Vue 又有着广泛的使用，所以将两者结合使用，是很多 Vue 项目开发者的切实需求。  
 2、目前没有发现满足这种需求，而使用又很方便的 repo、有的可能也只是简单的暴露一个 UEditor 的实例，仍然需要开发者手动去调用 getContent，setContent，而通过 v-model 绑定数据也是很多人期待的方式。于是自己在写公司项目时就手动撸了一个，周末整理一下分享出来，希望能帮助到有同样需求的小伙伴。

[](https://github.com/HaoChuan9421/vue-ueditor-wrap/raw/master/assets/images/demo.gif)

[点击预览](https://haochuan9421.github.io/vue-ueditor-wrap-demo/) [加入聊天室](https://gitter.im/haochuan9421/vue-ueditor-wrap/)

**Installation**

npm i vue-ueditor-wrap

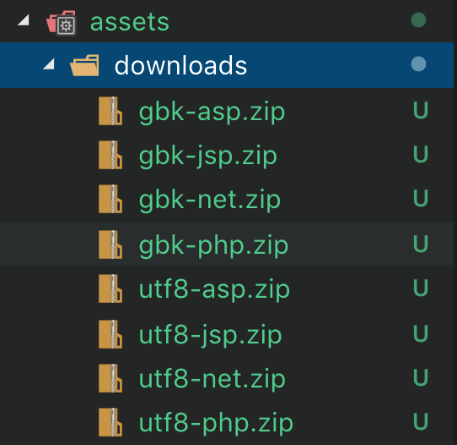
# 或者

yarn add vue-ueditor-wrap

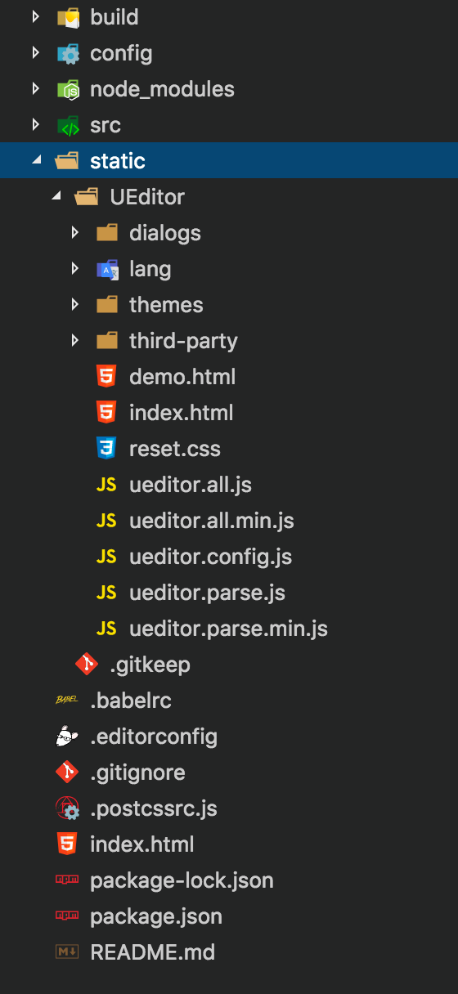
**Quick Start**

[基于 vue-cli 2.x 的完整 DEMO](https://github.com/HaoChuan9421/vue-ueditor-wrap-demo)  
[基于 Nuxt 的服务端渲染 DEMO](https://github.com/HaoChuan9421/vue-ueditor-wrap-nuxt)。

下载[最新编译的 UEditor](https://github.com/HaoChuan9421/vue-ueditor-wrap/tree/master/assets/downloads)。官网目前最新的版本是1.4.3.3，存在诸多 BUG，例如 [Issue1](https://github.com/HaoChuan9421/vue-ueditor-wrap/issues/1)，且官方不再积极维护。为了世界的和平，针对一些常见 BUG，我进行了[修复](https://github.com/HaoChuan9421/ueditor/commits/dev-1.4.3.3)，并把编译好的文件放在了本仓库的 assets/downloads 目录下，你可以放心[下载](https://github.com/HaoChuan9421/vue-ueditor-wrap/tree/master/assets/downloads)，当然你也可以自己 clone [官方源码](https://github.com/fex-team/ueditor)并[编译](http://fex.baidu.com/ueditor/#dev-bale_width_grunt)。

[](https://github.com/HaoChuan9421/vue-ueditor-wrap/raw/master/assets/images/downloads.png)

将下载的压缩包解压并重命名为 UEditor（只需要选择一个你需要的版本,比如 utf8-php）,放入你项目的 static 目录下。

[](https://github.com/HaoChuan9421/vue-ueditor-wrap/raw/master/assets/images/file.png)

如果你使用的是 [vue-cli 3.x](https://cli.vuejs.org/zh/guide/)，可以把 UEditor 文件夹放入项目的 public 目录下。

1. 引入VueUeditorWrap组件
2. import VueUeditorWrap from 'vue-ueditor-wrap' // ES6 Module
3. // 或者

const VueUeditorWrap = require('vue-ueditor-wrap') // CommonJS

你也可以通过直接引入 CDN 链接的方式来使用，它会暴露一个全局的 VueUeditorWrap 变量（具体如何使用你可以阅读我的这篇[博客](https://juejin.im/post/5b97b84ee51d450e6c7492f6)或参考这个[仓库](https://github.com/HaoChuan9421/vue-optimization/tree/cdn)）。

<script src="https://cdn.jsdelivr.net/npm/vue-ueditor-wrap@latest/lib/vue-ueditor-wrap.min.js"></script>

1. 注册组件
2. components: {
3. VueUeditorWrap
4. }
5. // 或者在 main.js 里将它注册为全局组件

Vue.component('vue-ueditor-wrap', VueUeditorWrap)

1. v-model绑定数据

<vue-ueditor-wrap v-model="msg"></vue-ueditor-wrap>

data () {

return {

msg: '<h2><img src="http://img.baidu.com/hi/jx2/j\_0003.gif"/>Vue + UEditor + v-model双向绑定</h2>'

}

}

至此你已经可以在页面中看到一个初始化之后的 UEditor 了，并且它已经成功和数据绑定了！👏👏👏

1. 根据项目需求修改配置，完整配置选项查看 [ueditor.config.js](https://github.com/HaoChuan9421/vue-ueditor-wrap/blob/master/public/UEditor/ueditor.config.js) 源码或 [官方文档](http://fex.baidu.com/ueditor/)

<vue-ueditor-wrap v-model="msg" :config="myConfig"></vue-ueditor-wrap>

data () {

return {

msg: '<h2><img src="http://img.baidu.com/hi/jx2/j\_0003.gif"/>Vue + UEditor + v-model双向绑定</h2>',

myConfig: {

// 编辑器不自动被内容撑高

autoHeightEnabled: false,

// 初始容器高度

initialFrameHeight: 240,

// 初始容器宽度

initialFrameWidth: '100%',

// 上传文件接口（这个地址是我为了方便各位体验文件上传功能搭建的临时接口，请勿在生产环境使用！！！）

serverUrl: 'http://35.201.165.105:8000/controller.php',

// UEditor 资源文件的存放路径，如果你使用的是 vue-cli 生成的项目，通常不需要设置该选项，vue-ueditor-wrap 会自动处理常见的情况，如果需要特殊配置，参考下方的常见问题2

UEDITOR\_HOME\_URL: '/static/UEditor/'

}

}

}

**Advanced**

1. 如何获取 UEditor 实例？

<vue-ueditor-wrap @ready="ready"></vue-ueditor-wrap>

methods: {

ready (editorInstance) {

console.log(`编辑器实例${editorInstance.key}: `, editorInstance)

}

}

1. 设置是否在组件的 beforeDestroy 钩子里销毁 UEditor 实例

<vue-ueditor-wrap :destroy="true"></vue-ueditor-wrap>

1. 选取 v-model 的实现方式。双向绑定的实现依赖对编辑器内容变化的监听，由于监听方式的不同，会带来监听效果的差异性，你可以自行选择，但建议使用开箱即用的默认值。

<vue-ueditor-wrap mode="listener"></vue-ueditor-wrap>

可选值：observer，listener

默认值：observer

参数说明：

* 1. observer 模式借助 [MutationObserver API](https://developer.mozilla.org/zh-CN/docs/Web/API/MutationObserver)。优点在于监听的准确性，缺点在于它会带来一点额外的性能开销。你可以通过 observerDebounceTime 属性设置触发间隔，还可以通过 observerOptions 属性有选择的设置 [MutationObserver](https://developer.mozilla.org/en-US/docs/Web/API/MutationObserverInit)的监听行为。该 API 只兼容到 IE11+，但 vue-ueditor-wrap 会在不支持的浏览器中自动启用 listener 模式。
  2. <vue-ueditor-wrap
  3. mode="observer"
  4. :observerDebounceTime="100"
  5. :observerOptions="{ attributes: true, characterData: true, childList: true, subtree: true }"
  6. >

</vue-ueditor-wrap>

* 1. listener 模式借助 UEditor 的 [contentChange 事件](https://ueditor.baidu.com/doc/#UE.Editor:contentChange)，优点在于依赖官方提供的事件 API，无需额外的性能消耗，兼容性更好，但缺点在于监听的准确性并不高，存在如下方 [常见问题 5] 中的提到的 BUG。

1. 是否支持 Vue SSR？

自 2.4.0 版本开始支持服务端渲染！本组件提供对 Nuxt 项目开箱即用的支持。但如果你是自己搭建的 Vue SSR 项目，你可能需要自行区分服务端和客户端环境并结合 forceInit 属性强制初始化编辑器，但大概率你用不到该属性，即使是自己搭建的 SSR 项目，更多问题欢迎提交 ISSUE。

1. 如何进行二次开发（添加自定义按钮、弹窗等）？

本组件提供了 beforeInit 钩子，它会在 UEditor 的 scripts 加载完毕之后、编辑器初始化之前触发，你可以在此时机，通过操作 window.UE 对象，来进行诸如添加自定义按钮、弹窗等的二次开发。beforeInit 的触发函数以 编辑器 id 和 配置参数 作为入参。下面提供了一个简单的自定义按钮和自定义弹窗的示例，[DEMO](https://github.com/HaoChuan9421/vue-ueditor-wrap-demo) 仓库中也提供了自定义“表格居中”按钮的示例，如果有更多二次开发的需求，你可以参考[官方 API](https://ueditor.baidu.com/doc/) 或者 [UEditor 源码](https://github.com/HaoChuan9421/ueditor/tree/dev-1.4.3.3/_examples) 中的示例。

自定义按钮 Demo自定义弹窗 Demo

**Features**

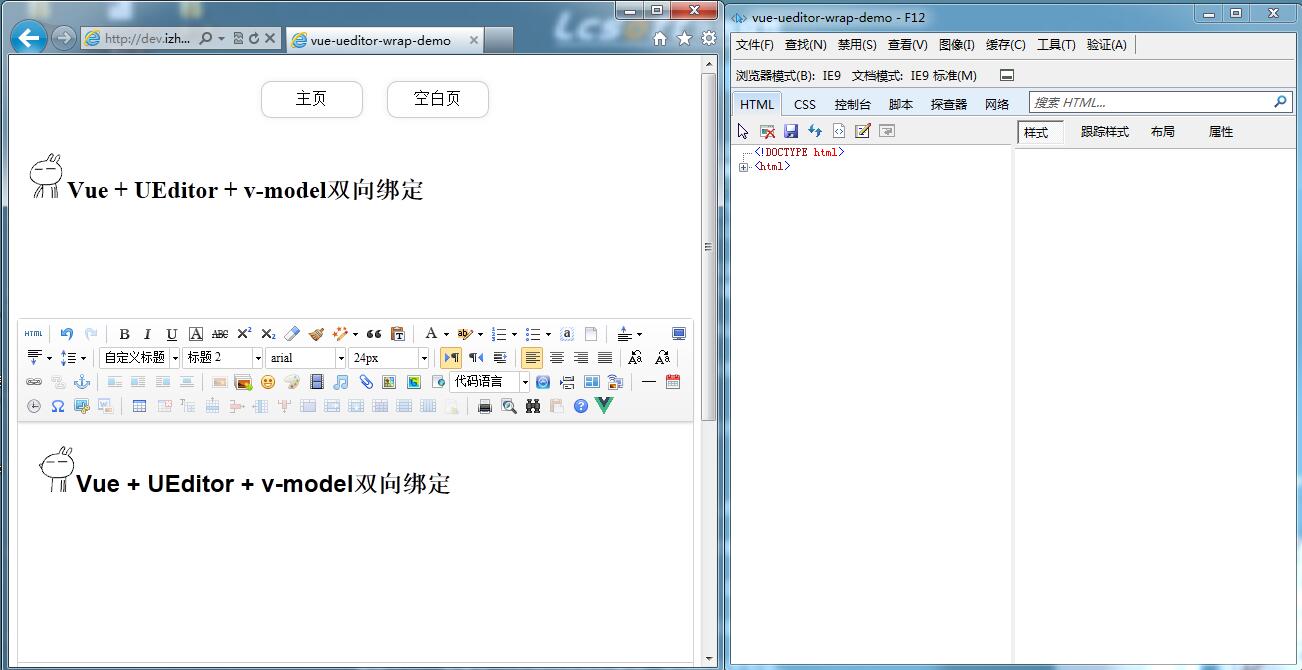
1. v-model 双向数据绑定！你不需要考虑实例化，也不需要考虑何时 getContent，何时setContent，简单到像使用 input框一样！
2. 完全遵从官方 API，所有的配置参数和实例方法与官方完全一致。通过给 vue-ueditor-wrap 组件的 config 属性传递一个对象，你就可以得到一个完全独立配置的 UEditor 编辑器。通过监听 ready 事件你就可以得到初始化后的 UEditor 实例并执行实例上的各种方法。
3. 自动添加依赖文件。你不需要自己在 index.html 或 main.js 里引入 UEditor 的 JS 文件。更重要的是即使你在一个页面里同时使用多个 vue-ueditor-wrap 组件，它所依赖的 JS 文件也只会加载一次。这么做的原因在于你不需要当用户一打开项目就先加载大量 UEditor 相关的资源，所有的资源文件只会在 vue-ueditor-wrap 组件第一次被激活时才加载。当然，如果你在 index.html 或 main.js 里引入了相关资源，vue-ueditor-wrap 也会准确判断，你不用担心它会重复加载。
4. 每个 vue-ueditor-wrap 组件是完全独立的。你甚至可以在上面使用 v-for 指令一次渲染 99个 兔斯基（不要忘记添加 key 值）。

[https://camo.githubusercontent.com/9be60d85e896f1dd9f33b6c18093e206733c065a/687474703a2f2f696d672e62616964752e636f6d2f68692f6a78322f6a5f303030332e676966](https://camo.githubusercontent.com/9be60d85e896f1dd9f33b6c18093e206733c065a/687474703a2f2f696d672e62616964752e636f6d2f68692f6a78322f6a5f303030332e676966)

**FAQ（常见问题）**

1. 是否支持 IE 等低版本浏览器？

与 Vue 相同，整体支持到 IE9+👏👏👏

[](https://github.com/HaoChuan9421/vue-ueditor-wrap/raw/master/assets/images/ie.jpg)

1. 为什么我会看到这个报错?

[https://github.com/HaoChuan9421/vue-ueditor-wrap/raw/master/assets/images/error1.png](https://github.com/HaoChuan9421/vue-ueditor-wrap/raw/master/assets/images/error1.png)

这是 UEDITOR\_HOME\_URL 参数配置错误导致的。在 vue cli 2.x 生成的项目中使用本组件，默认值是 '/static/UEditor/'，在 vue cli 3.x 生成的项目中，默认值是 process.env.BASE\_URL + 'UEditor/' 。但这并不能满足所有情况。例如你的项目不是部署在网站根目录下，如"http://www.example.com/my-app/"，你可能需要设置为"/my-app/static/UEditor/"。是否使用了相对路径、路由是否使用 history 模式、服务器配置是否正确等等都有可能会产生影响。总而言之：无论本地开发和部署到服务器，你所指定的 UEditor 资源文件是需要真实存在的，vue-ueditor-wrap 也会在 JS 加载失败时通过 console 输出它试图去加载的资源文件的完整路径，你可以借此分析如何填写。当需要区分环境时，你可以通过判断 process.env.NODE\_ENV 来分别设置。

1. 我该如何上传图片和文件？为什么我会看到后台配置项返回格式出错？

[https://github.com/HaoChuan9421/vue-ueditor-wrap/raw/master/assets/images/error2.png](https://github.com/HaoChuan9421/vue-ueditor-wrap/raw/master/assets/images/error2.png)

上传图片、文件等功能是需要与后台配合的，而你没有给 config 属性传递正确的 serverUrl ，我提供了http://35.201.165.105:8000/controller.php 的临时接口，你可以用于测试，**但切忌在生产环境使用！！！** 关于如何搭建上传接口，可以参考[官方文档](http://fex.baidu.com/ueditor/#server-deploy)。

1. 单图片跨域上传失败！

UEditor 的单图上传是通过 Form 表单 + iframe 的方式实现的，但由于同源策略的限制，父页面无法访问跨域 iframe 的文档内容，所以会出现单图片跨域上传失败的问题。我通过 XHR 重构了单图上传的方式，[下载最新编译的 UEditor](https://github.com/HaoChuan9421/vue-ueditor-wrap/tree/master/assets/downloads) 资源文件即可在 IE10+ 的浏览器中实现单图跨域上传了。具体细节，[点此查看](https://github.com/HaoChuan9421/ueditor/commit/31f9207142d21a406041da0bd97968b466530c76)。当然你也可以通过配置 toolbars 参数来隐藏单图片上传按钮，并结合上面介绍的“自定义按钮”，曲线救国，以下代码仅供参考。

var input = document.createElement('input')

input.type = "file"

input.style.display = 'none'

document.body.appendChild(input)

input.click()

input.addEventListener('change',(e)=>{

// 利用 AJAX 上传，上传成功之后销毁 DOM

console.log(e.target.files)

})

1. 为什么我输入的"? ! $ #" 这些特殊字符，没有成功绑定？

当你使用 listener 模式时，由于 v-model 的实现是基于对 UEditor 实例上 contentChange 事件的监听，而你输入这些特殊字符时通常是按住 shift 键的，UEditor 本身的 contentChange 在 shift 键按住时不会触发，你也可以尝试同时按下多个键，你会发现 contentChange 只触发一次。你可以使用 observer 模式或移步 [UEditor](https://github.com/fex-team/ueditor)。

1. 单图片上传后 v-model 绑定的是 loading 小图标。

这个也是 UEditor 的 BUG。我最新编辑的版本，修复了官方的这个 BUG，如果你使用的是官网下载的资源文件，请替换资源文件或参考 [Issue1](https://github.com/HaoChuan9421/vue-ueditor-wrap/issues/1)。

更多问题，欢迎提交 [ISSUE](https://github.com/HaoChuan9421/vue-ueditor-wrap/issues) 或者去 [聊天室](https://gitter.im/haochuan9421/vue-ueditor-wrap/) 提问。但由于这是一个个人维护的项目，我平时也有自己的工作，所以并不能保证及时解决你们的所有问题，如果小伙伴们有好的建议或更炫酷的操作，也欢迎 PR，如果你觉得这个组件给你的开发带来了实实在在的方便，也非常感谢你的Star，当然还有咖啡：

代码修改请遵循指定的 ESLint 规则，PR 之前请先执行 npm run lint 进行代码风格检测，大部分语法细节可以通过 npm run fix 修正，构建之后，记得修改 package.json 里的版本号，方便我 Review 通过后麻溜溜的发布到 npm。