single-spa-vue

single-spa-vue is a helper library that helps implement [single-spa registered application](https://single-spa.js.org/docs/configuration#registering-applications) [lifecycle functions](https://single-spa.js.org/docs/building-applications#registered-application-lifecycle) (bootstrap, mount and unmount) for for use with [Vue.js](https://vuejs.org/). Check out the [single-spa-vue github](https://github.com/single-spa/single-spa-vue).

single-spa-vue是一个帮助实现[single-spa注册应用程序](https://single-spa.js.org/docs/configuration#registering-applications)[生命周期函数](https://single-spa.js.org/docs/building-applications#registered-application-lifecycle)(初始化、挂载和卸载)的助手库，用于Vue.js。查看一下[single-spa-vue github](https://github.com/single-spa/single-spa-vue)。

##### 启动代码库

[#](https://single-spa.js.org/docs/ecosystem-vue#starter-repo)Starter repo

For a full example, see [coexisting-vue-microfrontends](https://github.com/joeldenning/coexisting-vue-microfrontends).

要获得完整的示例，请参见[coexisting-vue-microfrontend](https://github.com/joeldenning/coexisting-vue-microfrontends)。

##### 在线demo

[#](https://single-spa.js.org/docs/ecosystem-vue#live-demo)Live demo

[https://coexisting-vue-microfrontends.surge.sh](https://coexisting-vue-microfrontends.surge.sh/)

##### 安装说明

[#](https://single-spa.js.org/docs/ecosystem-vue#installation)Installation

###### vue CLI

[#](https://single-spa.js.org/docs/ecosystem-vue#vue-cli)Vue CLI

The [vue-cli-plugin-single-spa](https://github.com/single-spa/vue-cli-plugin-single-spa) will get everything set up.

[vue-cli-plugin-single-spa](https://github.com/single-spa/vue-cli-plugin-single-spa)会把一切都设置好。

vue add single-spa

The CLI Plugin does the following for you:

1. Modify your webpack config so that your project works as a single-spa application or parcel.
2. Install [single-spa-vue](https://github.com/single-spa/single-spa-vue).
3. Modify your `main.js` or `main.ts` file so that your project works as a single-spa application or parcel.
4. Add a `set-public-path.js` that will use `systemjs-webpack-interop` in order to set the public path of your application.

CLI插件为您做了以下工作:

1. 修改您的webpack配置，使您的项目作为single-spa应用程序或沙箱工作。

2. 安装single-spa-vue。

3.修改您的`main.js` 或 `main.ts`，使您的项目作为single-spa应用程序或沙箱工作。

4. 添加一个`set-public-path.js`,它将使用‘systemjs-webpack-interop’来设置应用程序的公共路径。

###### 不用vue CLI的方式

[#](https://single-spa.js.org/docs/ecosystem-vue#without-vue-cli)Without Vue CLI

npm install --save single-spa-vue

Alternatively, you can use single-spa-vue by adding

另外，您也可以通过添加来使用单spa-vue

<script src="https://unpkg.com/single-spa-vue"></script>

 to your html file and accessing the `singleSpaVue`  global variable.

到您的html文件，并访问' singleSpaVue '全局变量。

###### 使用方式

[#](https://single-spa.js.org/docs/ecosystem-vue#usage)Usage

Install `systemjs-webpack-interop` if you have not already done so.

如果你没有安装`systemjs-webpack-interop`，就按照下边的方式安装一下：

npm install systemjs-webpack-interop -S

Create a file at the same level as your `main.js/ts`  called `set-public-path.js`

创建一个与`main.js/ts` 相同级别的文件。文件名为“set-public-path.js”

import { setPublicPath } from 'systemjs-webpack-interop';

setPublicPath('appName');

Change your application's entry file to be the following.

按照下边的说明配置你应用的入口文件：

import './set-public-path';

import Vue from 'vue';

import App from './App.vue';

import router from './router';

import singleSpaVue from 'single-spa-vue';

const vueLifecycles = singleSpaVue({

Vue,

appOptions: {

render: h => h(App),

router,

},

});

export const bootstrap = vueLifecycles.bootstrap;

export const mount = vueLifecycles.mount;

export const unmount = vueLifecycles.unmount;

Note that if you are using the Vue CLI Plugin, your `main.ts` or `main.js` file will be updated with this code automatically and the `set-public-path.js` file will automatically be created with the app name being your package.json's name property.

If you want to deal with your Vue instance, you can modify the mount method by following this. mount method will return Promise with Vue instance after [v1.6.0](https://github.com/single-spa/single-spa-vue/releases/tag/v1.6.0).

注意，如果你使用的是Vue CLI插件，你的`main.ts` or `main.js`文件会自动更新这个代码和`set-public-path.js`文件将自动创建与应用程序的名称是package.json's的名字属性。如果希望处理Vue实例，可以按照下面的方法修改挂载方法。mount方法将在[v1.6.0](https://github.com/single-spa/single-spa-vue/releases/tag/v1.6.0)之后用Vue实例返回Promise。

const vueLifecycles = singleSpaVue({...})

export const mount = props => vueLifecycles.mount(props).then(instance => {

// do what you want with the Vue instance

...

})

##### 共享依赖

[#](https://single-spa.js.org/docs/ecosystem-vue#shared-dependencies)Shared dependencies

For performance, it is best to share a single version and instance of Vue, Vue Router, and other large libraries.

To do this, add your shared dependencies as [webpack externals](https://webpack.js.org/configuration/externals). Then you use an in-browser module loader such as [systemjs](https://github.com/systemjs/systemjs) to provide those shared dependencies to each of the single-spa applications. Adding `vue` and other libraries to your [import map](http://single-spa-playground.org/playground/import-map). For an example import map that is doing this, checkout [coexisting-vue-microfrontends' index.html file](https://github.com/joeldenning/coexisting-vue-microfrontends/blob/master/root-html-file/index.html).

Sharing a single instance of Vue and other common libraries is highly recommended. See the [recommended setup for single-spa](https://single-spa.js.org/docs/faq.html#is-there-a-recommended-setup) for more details on why.

为了提高性能，最好共享Vue、Vue路由器和其他大型库的单个版本和实例。

为此，将您的共享依赖项添加为[webpack扩展包](https://webpack.js.org/configuration/externals)。然后使用诸如systemjs之类的浏览器内模块加载器为每个单spa应用程序提供这些共享依赖项。将“vue”和其他库添加到 [import map](http://single-spa-playground.org/playground/import-map)中。对于这样的导入映射，请检出[oexisting-vue-microfrontends' index.html file](https://github.com/joeldenning/coexisting-vue-microfrontends/blob/master/root-html-file/index.html)文件。

强烈建议共享一个Vue实例和其他公共库。有关原因的更多细节，请参阅[single-spa的推荐设置](https://single-spa.js.org/docs/faq.html#is-there-a-recommended-setup)。

###### 用vue CLI 共享依赖

[#](https://single-spa.js.org/docs/ecosystem-vue#shared-deps-with-vue-cli)Shared deps with Vue CLI

// vue.config.js

module.exports = {

chainWebpack: config => {

config.externals(['vue', 'vue-router']);

},

};

###### 不用Vue CLI 共享依赖

[#](https://single-spa.js.org/docs/ecosystem-vue#shared-deps-without-vue-cli)Shared deps without Vue CLI

// webpack.config.js

module.exports = {

externals: ['vue', 'vue-router'],

};

###### 参数选项

[#](https://single-spa.js.org/docs/ecosystem-vue#options)Options

All options are passed to single-spa-vue via the `opts` parameter when calling `singleSpaVue(opts)`.The following options are available:

Vue: (required) The main Vue object, which is generally either exposed onto the window or is available via `require('vue')` `import Vue from 'vue'`

appOptions: (required) An object which will be used to instantiate your Vue.js application.  appOptions will pass directly through to `new Vue(appOptions)`

. Note that if you do not provide an `el`  to appOptions, that a div will be created and appended to the DOM as a default container for your Vue application.

loadRootComponent: (optional and replaces `appOptions.render`) A promise that resolves with your root component. This is useful for lazy loading.

To configure which dom element the single-spa application is mounted to, use [appOptions.el](https://vuejs.org/v2/api/#el):

当调用“singleSpaVue(opts)”时，所有选项都通过“opts”参数传递给single-spa-vue。以下是可供选择的方案:

Vue:(required)主要的Vue对象，它通常要么暴露在窗口上，要么通过' require(' Vue ') ' ' '从' Vue ' ' '导入Vue ' '可用。

appOptions:(必选)一个对象，将用于实例化Vue.js应用程序。appOptions将直接通过' new Vue(appOptions) '

。注意，如果不向应用程序提供“el”，那么将创建一个div并将其作为Vue应用程序的默认容器附加到DOM中。

loadRootComponent:(可选的，替换了' appOptions.render ')一个promise，它将用根组件解析。这对于延迟加载非常有用。

要配置single-spa应用程序要挂载到哪个dom元素，请使用[appOptions.el](https://vuejs.org/v2/api/#el):

const vueLifecycles = singleSpaVue({

Vue,

appOptions: {

render: h => h(App),

el: '#a-special-container',

},

});

##### 安装一个single-spa 沙箱

[#](https://single-spa.js.org/docs/ecosystem-vue#as-a-single-spa-parcel)As a single-spa parcel

To create a single-spa parcel, simply omit the `el` option from your appOptions, since the dom element will be specified by the user of the Parcel. Every other option should be provided exactly the same as in the example above.

要创建single-spa 沙箱，只需在应用程序中省略“el”选项，因为dom元素将由沙箱的用户指定。应该提供与上面示例完全相同的其他选项。

###### 用户属性

[#](https://single-spa.js.org/docs/ecosystem-vue#custom-props)Custom Props

[single-spa custom props](https://single-spa.js.org/docs/building-applications.html#custom-props) are added to your App component as `appOptions.data`, and are accessible via `vm.$data`. See [this Vue documentation](https://vuejs.org/v2/api/#data) for more information on `appOptions.data`.

[single-spa custom props](https://single-spa.js.org/docs/building-applications.html#custom-props) 作为 `appOptions.data`添加到应用程序组件中，并可通过“vm.$data”访问。有关“app .data”的更多信息，请参阅[Vue文档](https://vuejs.org/v2/api/#data)。