查看官方示例

1.git clone https://github.com/umijs/qiankun.git

2.cd qiankun

3.yarn install

4.yarn examples:install

5.yarn examples:start

第3步报错：

error eslint@5.8.0: The engine “node” is incompatible with this module. Expected version “^6.14.0 || ^8.10.0 || >=9.10.0”. Got “9.3.0”

解决：

[yarn](https://so.csdn.net/so/search?q=yarn&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/github_39088222/article/details/_blank) config set ignore-engines true 然后执行第3步

实战操作

**父应用配置**

1. 用Vue 随便搭建一个项目作为主应用，配置好路由，重点是添加vue-router
2. 安装乾坤依赖 yarn add qiankun # 或者 npm i qiankun -S
3. 在main.js中注册微应用并启动，完整代码为

import Vue from "vue";

import App from "./App.vue";

import router from "./router";

import store from "./store";

import { registerMicroApps, start } from "qiankun";

Vue.config.productionTip = false;

new Vue({

router,

store,

render: (h) => h(App),

}).$mount("#app");

// 在主应用中注册子应用

registerMicroApps([

{

name: "vueApp", //子应用名称

entry: "//localhost:8091", //子应用启动的地址

container: "#container", // 子应用在主应用的容器名称

activeRule: "/app-vue", // 路由地址

props: {

data: "child子应用",

}, //传参

},

]);

// 启动

start();

1. 配置主应用跨域，在项目根目录下创建vue.config.js文件，并添加如下代码

module.exports = {

devServer: {

port: 8090, //主应用启动端口号，视情况而定

headers: {

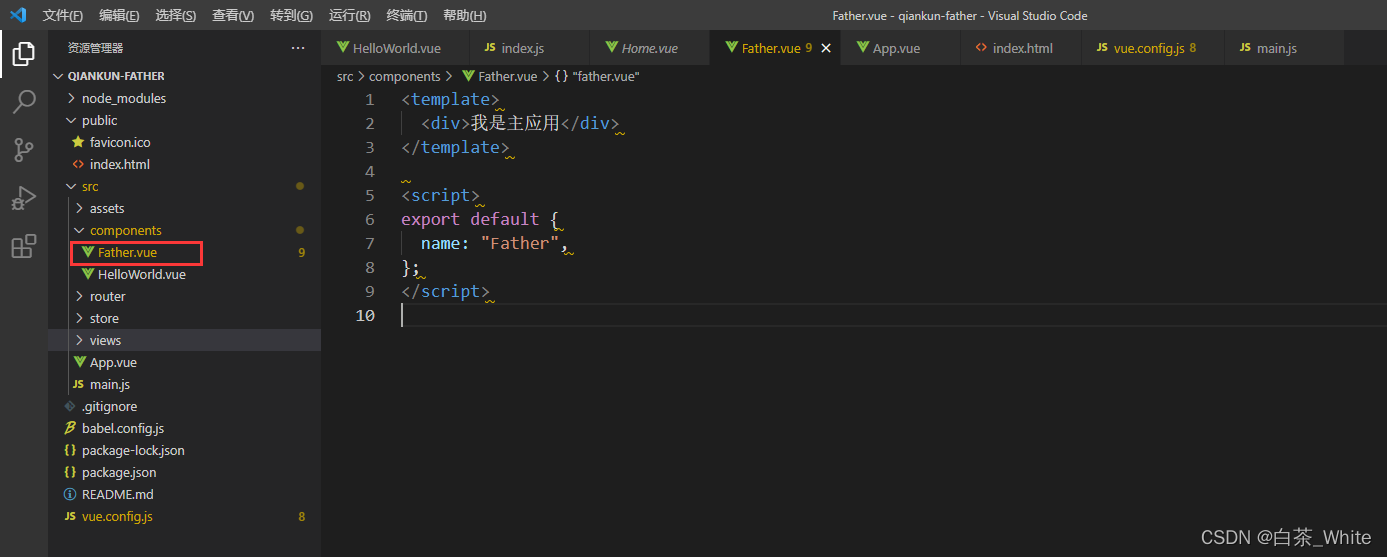
"Access-Control-Allow-Origin": "\*", // 允许跨域访问子应用页面

},

},

};

1. 新建父页面，在src/components文件夹下新建父页面，Father.vue,代码及文件位置如下图所示



1. 配置路由，打开src/router文件夹下的index.js文件，配置Father页面的路由，完整代码如下图所示

import Vue from "vue";

import VueRouter from "vue-router";

Vue.use(VueRouter);

const routes = [

{

path: "/",

redirect: "/father",

},

{

path: "/father",

name: "Father",

component: () => import("@/components/Father"),

},

];

const router = new VueRouter({

mode: "history",

routes,

});

export default router;

1. 配置App.vue,仅修改HTML部分代码即可，代码如下

<div id="app">

<span><router-link to="/">点击跳转到父页面 </router-link></span>

<span><router-link to="/app-vue">点击跳转到子页</router-link></span>

<router-view />

<div id="container">

<!-- 子应用位置 -->

</div>

</div>

1. npm-run-all用来通过执行一条语句来达到执行多条语句的效果的插件

npm install npm-run-all --save-dev或者yarn add npm-run-all --dev

再改造package.json后执行npm run serve

 "scripts": {

    "serve": "npm-run-all --parallel serve:\*",

    "serve:box": "vue-cli-service serve",

    "serve:web1": "cd subapp/web1 && yarn serve",

    "serve:web2": "cd subapp/web2 && yarn serve",

    "build": "npm-run-all --parallel build:\*",

    "build:box": "vue-cli-service build",

    "build:web1": "cd subapp/web1 && yarn build",

    "build:web2": "cd subapp/web2 && yarn build"

  },

**子应用配置**

1. 使用vue脚手架新建一个vue项目，重点是添加vue-router
2. 在src文件夹下新建public-path.js文件，并添加如下代码

if (window.\_\_POWERED\_BY\_QIANKUN\_\_) {

\_\_webpack\_public\_path\_\_=window.\_\_INJECTED\_PUBLIC\_PATH\_BY\_QIANKUN\_\_;

}

如报错webpack\_public\_path 未定义，则在package.json中的eslintConfig中进行 如下配置"eslintConfig": {

......

"globals": {

"\_\_webpack\_public\_path\_\_": true

}

},

1. 修改main.js代码，完成代码如下

import "./public-path";

import Vue from "vue";

import App from "./App.vue";

import router from "./router";

Vue.config.productionTip = false;

let instance = null;

function render(props = {}) {

const { container } = props;

instance = new Vue({

router,

render: (h) => h(App),

}).$mount(container ? container.querySelector("#app-vue") : "#app-vue");

}

// 独立运行时

if (!window.\_\_POWERED\_BY\_QIANKUN\_\_) {

render();

}

export async function bootstrap() {

console.log("[vue] vue app bootstraped");

}

export async function mount(props) {

console.log("[vue] props from main framework", props);

render(props);

}

export async function unmount() {

instance.$destroy();

instance.$el.innerHTML = "";

instance = null;

}

new Vue({

router,

render: (h) => h(App),

}).$mount("#app-vue");

与官网文档差异import routes from "./router";==>import router from "./router";

注意修改了#app为#app-vue，原因为#app在主应用中已使用，会冲突，所以对应的index.html与App.vue中也应当修改为#app-vue

1. 配置子应用跨域，在项目根目录下创建vue.config.js文件，并添加如下代码

const { name } = require("./package");

module.exports = {

devServer: {

port: 8091, //子应用启动端口号，不可随意修改，与上文中父应用注册的子应用 端口号对应

headers: {

"Access-Control-Allow-Origin": "\*", //循序跨域

},

},

configureWebpack: {

output: {

library: `${name}-[name]`,

libraryTarget: "umd", // 把微应用打包成 umd 库格式

jsonpFunction: `webpackJsonp\_${name}`,

},

},

};

1. 配置路由，打开src/router文件夹下的index.js文件，将以下代码

const router = new VueRouter({

routes,

});

修改为

const router = new VueRouter({

// 这里和主应用中注册子应用时的activeRule对应

base: window.\_\_POWERED\_BY\_QIANKUN\_\_ ? "/app-vue" : "/",

mode: "history",

routes,

});

1. 配置App.vue,仅修改HTML部分代码即可，代码如下

<template>

<div id="app-vue">我是子应用</div>

</template>

注意，这里的id对应了main.js中的名称，不要再使用id="app",记得同步 修改index.html中的id

路由配置

在微应用的路由配置文件中，需要增加主应用中配置时对应的路由根路径，同时区分是否为微服务，微服务情况下添加路由前缀，非微服务不需要添加。

1 src/router.js

2 const prefix = window.\_\_POWERED\_BY\_QIANKUN\_\_ ? '/home/micro-fronted-query' : ''

3 const appRouters = [

4 {

5 path: `${prefix}/`,

6 redirect: `${prefix}/login`

7 },

8 {

9 path: `${prefix}/login`,

10 name: 'login',

11 meta: {

12 title: 'Login - 登录'

13 },

14 component: () => import('./views/login.vue')

15 },

16 {

17 path: `${prefix}/home`,

18 component: () => import('./views/home.vue')

19 },

20 {

21 path: `${prefix}/list`,

22 component: () => import('./views/list.vue')

23 }

24 ]

25

26 export default appRouters