VUE.JS笔记

[搭建脚手架 1](#_Toc25927)

[全局注册组件 2](#_Toc16715)

[局部注册组件 2](#_Toc7898)

[传值和传引用 3](#_Toc32588)

[子向父组件传值 4](#_Toc18631)

[生命周期（钩子函数） 6](#_Toc26964)

[路由 8](#_Toc6242)

[http请求接口 10](#_Toc521)

[Vuex的文档 11](#_Toc9685)

[管理后台的示例 13](#_Toc2205)

# 搭建脚手架

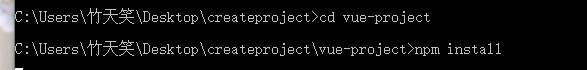
步骤一：# 全局安装 vue-cli

$ npm install --global vue-cli

输入vue --version出现版本号是安装过了的







1. vue init webpack vue-project
2. Cd vue-project
3. Npm install
4. Npm run dev

最后输入npm run dev搭建完毕。

# 全局注册组件

Main.js

import Users from './components/Users'

Vue.component("users",Users);

# 局部注册组件

App.vue

<script>

import Users from './components/Users'

export default {

name: 'app',

data(){

return{

title:'这是我的第一个脚手架项目'

}

},

components:{

"users":Users(或者直接写Users)

}

}

</script>

在components里面注册一个vue，然后在app.vue里面局部一定一个组件就可以用了

# 传值和传引用

//props:['a'],//拿到app.vue里面绑定的a

props:{

a:{

type:String，

required:true

}

}

传值：string nnumber bolean( 只会改变自己的值)

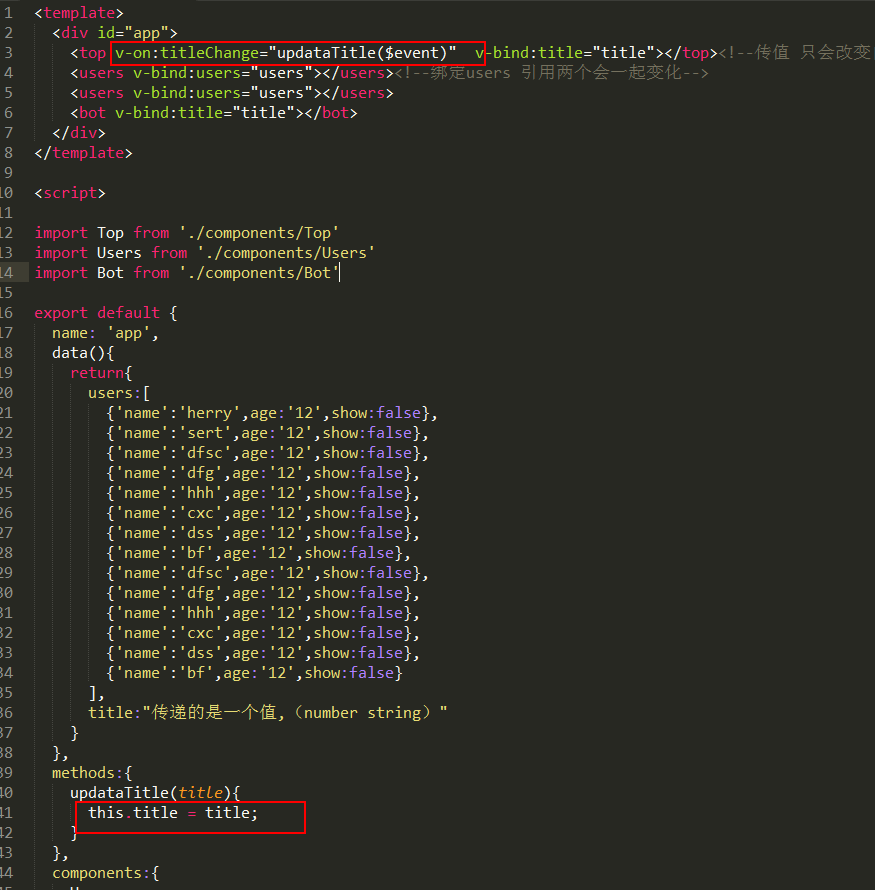
引用: array object(改变一个地方的数据 跟这个引用相关的都会变化)

# 子向父组件传值

子组件中先注册一个事件



在父组件中找到titleChange----updataTitle----改变title



# 生命周期（钩子函数）

beforeCreate:function(){

alert('组件实例化之前执行的函数')

},

create:function(){

alert('组件实例化完毕，但页面还未显示')

},

beforeMount:function(){

alert('组件挂在前，页面任未展示，但是虚拟dom已经配置')

},

mounted:function(){

alert('组件挂在后，此方法执行后，页面显示')

},

beforeUpdata:function(){

alert('组件更新前,页面仍未更新，但虚拟dom已经配置')

},

updated:function(){

alert('组件更新,此方法执行后，页面显示')

},

beforeDestroy:function(){

alert('组件销毁前')

},

destroyed:function(){

alert('组件销毁')

}

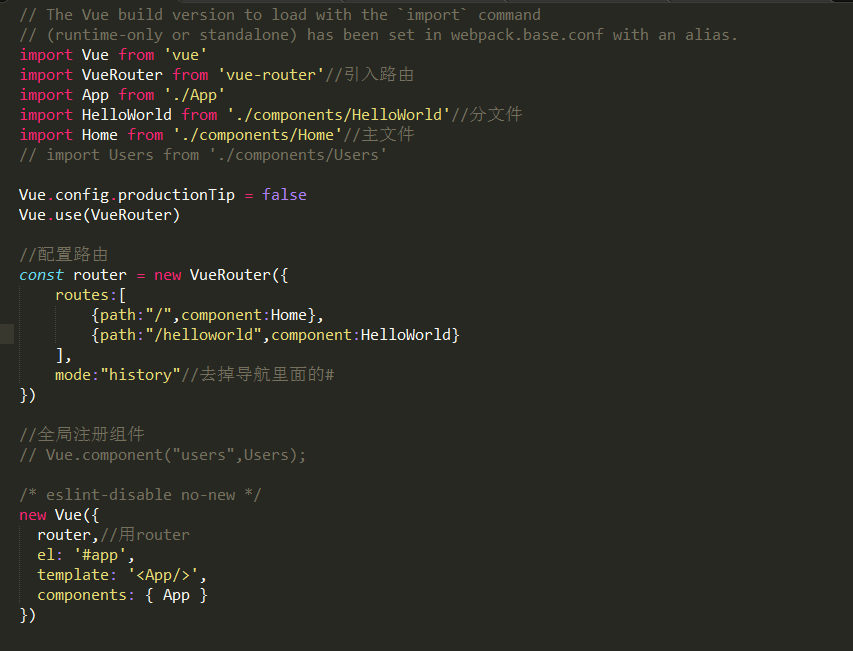
# 路由

安装路由

npm install vue-router --save-dev

安装后重启

在main.js中引入路由，引入文件



//配置路由

const router = new VueRouter({

routes:[

{path:"/",component:Home},

{path:"/helloworld",component:HelloWorld}

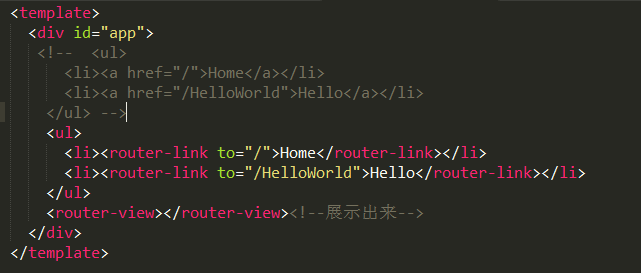
],

mode:"history"//去掉导航里面的#

})

用路由设置跳转链接

A链接会重新刷新页面，router-link 不会重刷页面。



# http请求接口

安装vue resource(命令：npm install vue-resource --save-dev)

在main.js中引入以及使用VueResource 。

import VueResource from 'vue-resource'

Vue.use(VueResource)

请求接口使用created函数

created(){

this.$http.get("http://jsonplaceholder.typicode.com/users"). then((data) => {

this.users = data.body;

})

}

将请求的data传到users 里面显示出来。

# Vuex的文档

安装vuex：npm install vuex --save

State:用于存储数据

Getters:获取数据

Mutations:触发事件变更状态的时候

Action :可以包含任意异步操作，点击事件之后的延时可以用这个

若Getters、actions里面有很多方法直接用：

import {mapGetters} from 'vuex'（用之前先引用过来）

...mapGetters([

"saleProducts"

])

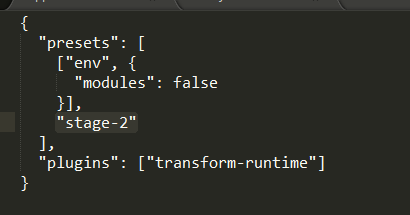
...mapActions([

"reducePrice"

])。。。

若是报错就是不识别es6语法要安装一下，并且在babelrc里面添加"stage-2"。

安装es6语法的东西：npm install babel-preset-stage-2 --save-dev



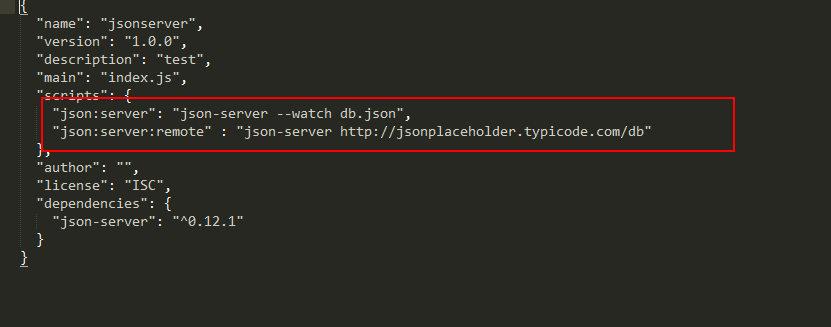
# 管理后台的示例

1. 安装json-server(npm install -g json-server),全局安装。
2. 进入JSONSERVER文件夹npm init 下载对应的插件
3. 局部安装进入文件夹安装：npm install json-server --save

4.自己创建一个json文件写json数据，

在package.json里面的script写：

"json:server": "json-server --watch db.json"



在cmd里面启动json:server( npm run json:server )

搭建一个脚手架。（npm init webpack 项目名 -- cd 项目名 -- npm install -- npm run dev）

1. 开始写项目

创建一个Custorm.vue和About.vue文件，有路由的安装路由，main.js中引入路由、设置路由。

1. Index文件中引入bootstrapjs和css方便编写，在main.js的new Vue里面引入bootstrap的结构
2. 在custorm编写表格结构，获取json数据，



1. 添加用户

新建各个所需的vue，在