# 前端框架对比 vue angular react

1. 单页面框架
2. 基于模块化的开发模式

# Vue安装

1. 搭建环境，必须安装node.js，
2. 安装Vue脚手架， npm install --global vue-cli
3. 创建项目

vue init webpack vueproject(创建项目)

cd vueproject （进入项目文件路径）

cnpm install /npm install(安装依赖）

npm run dev (运行项目）

Eslint 是代码检查工具，不建议安装;cnpm就是淘宝镜像，如果npm安装不成功，就安装cnpm,较快。

另一种简单创建项目方式：

vue init webpack-simple vueproject

cd vueproject

cnpm install/npm insatll

npm run dev

项目文件目录

拿到一个新的项目需加载运行cnpm install/npm install,安装依赖包，生成modelues文件。package.json是管理项目的文件，创建项目的一些信息，webpack.config是webpack的配置文件。

APP.vue是根组件。

Main.js是挂载组件、vue对象。

组件介绍，结构如下：

<template>

  <div id="app"></div>

</template>

<script>

export default {

  name: "app",

  data() {

    return {};

  }

};

</script>

<style>

</style>

组件模板中的内容必须包含在一个根节点中，比如，模板中必须添加一个div或者其他标签。

所有内容放在该根节点中。Template中是html代码，主要就是放在一个根节点中。<script>标签中放的是业务逻辑，固定语法是

export default {

  name: "app",

  data() {

    return {};

  }

};

目的就是把组件暴露出去供页面使用。

Name：‘app’,就是组件的名称，可以删除。Data中是业务逻辑需要处理的数据，用{{}}

取值，data返回值必须用return{}方式返回出去。

数据绑定，数据渲染

Vue绑定属性，绑定css,绑定style，

通过v-bind把属性和值进行绑定，全写是v-bind：data= ‘’111 ‘’ 简写是：data= ‘’1111 ‘’

绑定html是v-html，绑定数据也可以是v-text，v-text可以简写为{{}}

vue的双向数据绑定，及vue中的ref获取dom节点：

V-on:click 绑定点击事件，简写为@click,v-model,

vue中给标签元素定于ref，也就是类dom元素的id，

Vue中获取dom节点的方式为this.$ref.userinfo

ref获取dom节点

改变样式的颜色：this.$refs.box.style.background = ‘ red’

vue事件：

v-on :click = ‘ run()’====>@click = ‘ run()’

V-bind:src = ‘url’ =====>:src = ‘url’

methods:{

Run:function(){}

Run2(){}//es6写法，不用function

}

Vue方法中获取数据用this.msg

微信小程序使用this.data.msg

react中用this.state.msg

angular中使用this.msg

事件传值中，传递事件对象

<button @click="eventinfo($event)">事件对象</button>

  e 为dom节点，e.srcElement 获取dom节点

  也可以利用事件对象获取定义属性

  比如<button data-aid = '111111' @click="eventinfo($event)">事件对象</button>

  获取自定义属性为e.srcElement.dataset.aid 获取自定义传值

  键盘enter用keycode ==13获取

  @keydown = '$event'获取事件对象

  增加缓存，localStorage.setItem('key',JSON.stringify(this.lis))

  在生命周期钩子函数中加载时获取mounted,

  例如：

  mounted（）{

      //生命周期函数，会在页面刷新时触发

  }

该函数与methods同级别

change方法类似click使用，例如@change  = ‘add()’数值改变时，会触发这个方法

对象都有set和get方法，比如在文件夹中新建一个model文件夹，然后新建一个storage.js文件

var starge = {

    set(key,value){

       localStorage.setItem('key',JSON.stringify(value));

    },

    get(key){

        return JSON.parse(localStorage.getItem('key'))

        remove(key){

           localStorage.removeItem('key')

        }

    }

}

把模块暴露出去export.defulat storage；

./表示当前目录

在.vue文件中进行引用。

在<script>标签中写import storage from './model/storage.js'

    组件，组件就是把公共部分抽成组件

    组件由模板，script,css组成，

    使用组件的三步骤：

    1、引入组件，引入组件必须在<script>标签中，import....

    2、挂载组件，必须在components中注册挂载

    3、模板中使用，在模板的根dom中写入<组件名称></组件名称>

    vue 中所以的模板必须有一个根节点

    组件添加局部样式用<<style lang="stylus" scoped>

    </style>

生命周期函数：组件挂载、更新、销毁、时触发的方法，组件销毁时调用的方法是destoryed(),生命周期的方法有：beforeCreate，created，

beforeMount,mounted,beforeUpdate,updated,beforeDestory,destoryed

请求数据的方法：vue-resource，axious,fetch-jsonp

1、请求数据模板，vue-resource是官方提供的一个插件，使用方法为

            1-cnpm install vue-rescourse --save(save是保存在package.json中)。

            2-在main.js中引入 import VuerRescoure from  'vue-rescourse',

            3-使用插件：vue.use(VuerRescoure),

            4、this.$http.get(地址).then(function(response){})

            var api = "http://www.phonegap100.com/appapi.php?a=getPortalList&catid=20&page=1";

        this.$http.get(api).then(function (response) {

          console.log(response);

          this.list = response.body.result;

        },function (err) {

          console.log(err);

        })

            // 传统写法

            this.$http.get('/someUrl', [options]).then(function(response){

                // 响应成功回调

            }, function(response){

                // 响应错误回调

            });

            // Lambda写法

            this.$http.get('/someUrl', [options]).then((response) => {

                // 响应成功回调

            }, (response) => {

                // 响应错误回调

            }

在生命周期中调用函数也是使用this.方法名（）的方式进行调用，也就是直接用this调用

2、axios,是第三方插件，

            1、cnpm install axios --save

            2、哪里使用就在哪里引入。比如在home.vue中使用，就在home.vue中引入import Axiox from 'axios',

            3、方法中引用：

            getData(){

                var api = 'url'

                Axios.get (api).then(function(){}).catch(function(){})

                或者用箭头函数表示

                Axios.get.get(api).then((res)=>{console.log(res)}).catch((error)=>{console.error(error)})

            }

3、JSONP同axios的用法一样

路由嵌套：

1、配置路由

        1、在json数据中加入路由配置

        2、在main.js中也将路由加入 import router from './router'

        3、最后要在router文件夹中index.js中配置路由

            path中填具体路径

            component中填跳转页面的位置

            import Vue from 'vue';

            import Router from 'vue-router';

            Vue.use(Router);

            export default new Router({

            mode: "history",

            routes: [

                {

                //首页组件

                path: '/',

                redirect: '/index',

                component: resolve => require(['../components/common/Home.vue'], resolve),

                meta: {title: '首页'},

                children: [

                    {

                    path: '/index',

                    component: resolve => require(['../components/page/Index.vue'], resolve),

                    meta: {title: '关注'}

                    },

                    {

                    //关注组件

                    path: '/subscriptions',

                    component: resolve => require(['../components/page/Subscriptions.vue'], resolve),

                    meta: {title: '关注'}

                    },   ]

                },

                {

                // 写文章

                path: '/write',

                component: resolve => require(['../components/page/Write.vue'], resolve),

                meta: {title: '写文章'}

                }

            }]

            })

2、父路由配置子路由显示的位置<router-view></router-view>

vuex是用来不同组件进行传值--------store.js

1、在src目录下新建一个vuex的文件夹

2、vuex中新建一个store.js

3、安装vuex, cnpm install vuex --save

4、在刚创建的store.js文件中引入vue，同时引入vuex，并且使用vuex，

    import Vue from 'Vue'

    import Vuex from 'Vuex'

5、定义数据 var state = {//state用来存储数据

                count ：1

            }

6、定义方法：mutations里面放的是方法，方法主要是用来改变state中的数据

            var mutations = {

                incCount(){

                    ++state.count;

                }

            }

7、暴露store供其他组件或者页面调用

            const store = new Vuex.Store({

                    state,

                    mutations

            })

            export default store

在组件中使用store.js

 1、引入store，import store from '......'

 2、注册store，和data(){}平级

    比如：

    data（）{}

    store，

    methods:{}

3、使用store，在模板中使用为{{store.state.count }}

4、改变store中的数值方式为在方法中写入this.$store.commit('inCount');//inCount是store中的方法

vuex是用来不同组件进行传值-----------------vue 通过属性访问，实现不同组件的数据共享

在store.js中

            var state = {//state用来存储数据

                count ：1

            }

            var mutations = {

                incCount(){

                    ++state.count;

                }

            }

            var getters = {//getter方法类似计算属性，在修改store中的count时，会触发getters中的方法并获取新的值

                    computedCounted{

                        (store)=>{

                            return state.count\*2

                        }

                    }

            }

            同时在store中进行暴露getters

            const store = new Vuex.Store({

                    state,

                    getters,

                    mutations

            })

            export default store

在组建中通过this.$store.getters.computedCounted调用触发属性值的改变事件方法。

通过Action 不同组件数据共享

Action 类似mutation，Action提交的是mutation而不是直接更改状态，Action可以包含异步操作

用法为在store.js中写

             var actions = {//getter方法类似计算属性，在修改store中的count时，会触发getters中的方法并获取新的值

                    inMutationsCount(context){

                        //通过调用context.commit提交一个mutation,执行了mutation中的inCount方法，改变state中的数据

                        context.commit（'inCount'）

                    }

            }

在组件中触发mutations中的方法用commit，触发actions中的方法用this.$store.dispatch('inMutationsCount');

最后记得在store.js中把actions暴露出来，也就是

            const store = new Vuex.Store({

                    state,

                    getters,

                    actions，

                    mutations

            })

            export default store

父子组件传值:

    1、父组件调用子组件时，绑定属性

    2、子组件用props:[]接收父组件传的值，如果子父组件的属性名一样，则取父组件

父子组件传方法：

    1、父组件写方法，

    2、调用子组件绑定方法属性

    3、子组件用props:[]接收方法

父子组件传父组件

    1、将父组件绑定到this,子组件用props获取

    验证父组件的传值得合法性

    props：{

        'title':String

    }

不验证的话就是props：[]

父组件主动获取子组件的数据和方法

    1、调用组件时，定义一个ref,例如<v-header ref = 'header'></v-header>

    2、在父组件中通过this.$refs.header.属性名 获取属性值，通过this.$refs.header.方法名 调用方法

子组件主动获取父组件的数据和方法

    1、this.$parent.属性名

    2、this.$parent.方法名

vue 非子父组件进行传值，比如非父子组件或者兄弟组件传值

1、新建js文件，引入vue实例（参考app.vue),最后暴露出来

2、在要广播的地方插入刚定义的实例，方法为VueEmit.$emit('名称'，数据)

3、在接收数据的地方通过VueEmit.$on('名称'，function(data){.......})处理数据

路由：vue路由配置

        1、安装路由  cnpm install vue-router --save

        2、在main.js中引用并使用vue-router

            import VueRouter from 'vue-router';

            Vue.use(Router);

        3、配置路由 在main.js中写入

                    1、创建组件，引入组件

                    2、定义路由

                    const routers= [

                        {path:'/foo',component:Foo}

                    ]

        4、实例化VueRouter

        const router = new VueRouter({

            routes //缩写，相当于routes：routes

        })

        5、挂载

        new Vue({

            el:'#app',

            router,

            render:h=>h(App)

        })

        6、在根组件模板中App.vue中放入<router-view></router-view>,或者在其他需要显示的组件中添加view标签，动态加载

        路由的重定向地址，默认打开到特定地址：

        {path:'\*',redirect:'/path'}

        动态路由传值 mounted(){

            this.$router.params

        }

动态路由路径后面的书写形式为/content/:id,则在组建中的<router-link to = '路径'>标签中的路径后面跟/路径/key

    动态路由的配置：path:/user/:id

    在对应页面获取动态路径的值方式为this.$route.params

    get传值：

    1、同路由动态传值

    配置path：'/user'无后缀

    2、在router-link中的to属性后面陪path?id = key

    3、get方式取值this.$route.query

    如果是jsonp()请求，后台接口需要支持jsonp

    比如：this.$http.jsonp(api).then(function(res){console.log(res)},funcction(error){console.error(error)})

    引入公共的上厕所时，创建项目是必须是上scss,在main.js中引入  import './assets/css/basicscss'

    编程式导航，就是页面跳转，利用JavaScript来实现页面跳转，方法为  this.$router.push({path:'news'})

    更改history模式就是在路由声明时，把mode:'history'改成hash，在main.js中