

link: null
title: 珠峰架构师成长计划
description: 在main.js引入iview
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats paragraph=50 sentences=101, words=854

创建项目

```
vue create <project>  
cd project  
yarn install axios iview  
</project>
```

使用iview

在main.js引入iview

```
import iView from 'iview';  
import 'iview/dist/styles/iview.css';  
Vue.use(iView);
```

配置axios参数以及拦截器

对请求参数拦截器进行封装

```
import axios from 'axios';  
import store from '../store';  
import {getLocal} from '../libs/local'  
class AjaxRequest{  
  constructor(){ //  6#x914D;6#x7F6E;6#x57FA;6#x672C;6#x53C2;6#x6570;  
    this.baseUrl = process.NODE_ENV === 'production'?/': 'http://localhost:3000'  
    this.timeout = 3000;  
    this.queue = {};  
  }  
  merge(options){  
    return {...options,baseUrl:this.baseUrl,timeout:this.timeout}  
  }  
  setInterceptor(instance,url){  
    //  6#x8BF7;6#x6C42;6#x524D;  6#x6267;6#x884C;6#x6B64;6#x65B9;6#x6CD5;  
    instance.interceptors.request.use(config=>{  
      config.headers.authorization = getLocal('token');  
      if(Object.keys(this.queue).length === 0){  
        store.commit('show_loading')  
      }  
      this.queue[url] = url;  
      return config;  
    });  
    //  6#x8BF7;6#x6C42;6#x540E;6#x6267;6#x884C;6#x6B64;6#x65B9;6#x6CD5;  
    instance.interceptors.response.use(res=>{  
      store.commit('hide_loading')  
      delete this.queue[url];  
      if( Object.keys(this.queue).length === 0){  
        store.commit('hide_loading')  
      }  
      return res.data;  
    });  
  }  
  request(options){  
    let instance = axios.create();  
    let config = this.merge(options);  
    this.setIntercepotor(instance,options.url);  
    return instance(config); // axios();  
  }  
}  
export default new AjaxRequest();
```

配置vuex存储用户状态

用户信息通过vuex来进行存储

```
import {login, validate} from './api/user'
import {setLocal} from './libs/local'
export default new Vuex.Store({
  state: {
    username: '', // 5B58;50A8;7528;6237;
    isShowLoading: false // 8C03;7528;63A5;53E3;65F6;662F;5426;663E;793A; loading
  },
  mutations: {
    hide_loading(state) {
      state.isShowLoading = false;
    },
    show_loading(state) {
      state.isShowLoading = true;
    },
    set_user(state, payload) {
      state.username = payload
    }
  },
  actions: {
    async set_user({commit}, username) {
      let data = await login(username)
      if (data.code !== 0) {
        return Promise.reject(data.data);
      }
      commit('set_user', data.username);
      setLocal('token', data.token);
    },
    async validate({commit}) {
      let r = await validate();
      if (r.code === 0) {
        commit('set_user', r.username);
        setLocal('token', r.token);
      }
      return r.code === 0;
    }
  }
})
```

全局拦截器配置

页面切换时进行权限获取

```
router.beforeEach(async (to, from, next) => {
  let needLogin = to.matched.some(match => match.meta.needLogin);
  let result = await store.dispatch('validate');
  if (needLogin) { // 9700;8981;767B;5F55;
    if (result) {
      next();
    } else {
      next({path: '/login'})
    }
  } else { // 4E0D;9700;8981;767B;5F55;
    if (result && to.path === '/login') { // 5DF2;7ECF;767B;5F55;800C;4E14;8FD8;662F; login
      next({path: '/'})
    } else {
      next();
    }
  }
});
```

EventBus应用

```
// Boy 7EC4;4EF6; 53D1;5C04; dinner 4E8B;4EF6;
<template>
  <div>5737;5B69;
    <button @click="sayToGirl()">5BF9;5973;5B69;5BFF4;58BDD;</button>
  </div>
</template>
<script>
export default {
  methods: {
    sayToGirl() {
      this.$bus.$emit('dinner', '你饿吗');
    }
  }
}
</script>
```

```
// Girl 7EC4;4EF6; 76D1;542C; dinner 4E8B;4EF6;
<template>
  <div>
    5973;5B69; <span>5737;5B69;5BF9;6211;5BFF4;5FF1A; {{message}}</span>
  </div>
</template>

<script>
import { default }
export default {
  data() {
    return { message: '' }
  },
  mounted() {
    this.$bus.$on('dinner', (data) => {
      this.message = data
    })
  }
}
```

render函数

- 第一个参数标签
- attrs, style, props, on, nativeOn, class, directives, slot, key, ref, domProps
- 第三个参数内容

渲染dom元素

```
h('div',{
  class:'test',
  attrs:{
    id:1
  },
  style:{
    color:'red'
  },
  on:{
    click(){
      alert(1)
    }
  }
},'hello world');
```

渲染组件

```
h(Test,{
  props:{
    name:'zf'
  },
  on:{
    change(){
      alert('123')
    }
  },
  nativeOn:{
    click(){
      alert(1);
    }
  }
},'hello world')
```

render函数优势

```
Vue.component('MyTitle',{
  props:{
    level:{},
  },
  render(h){
    return h('h'+this._self.level,this.$slots.default)
  }
});
new Vue({
  router,
  store,
  render:h=>{
    return h('MyTitle',{
      props:{
        level:2
      }
    },'hello world')
  }
}).$mount('#app');
```

用jsx改写

```
Vue.component('MyTitle',{
  props:{
    level:{},
  },
  render(h){
    let tag = 'h'+this.level;
    return <tag>{this.$slots.default}</tag>
  }
});
new Vue({
  router,
  store,
  render:h=>{
    return <mytitle level="1">hello world</mytitle>
  }
}).$mount('#app');
```

函数式组件

通过render函数订制组件

```

<list :render="renderFn"></list>
import List from './components/List.vue';
export default {
  components:{
    List
  },
  methods:{
    renderFn: (h,name)=>{
      return <h1>{name}</h1>
    }
  }
}

// 0x901A;0x8FC7;0x51FD;0x6570;0x5F0F;0x7EC4;0x4EF6;0x8FDB;0x884C;0x6570;0x636E;0x7684;0x8F6C;0x5316;
<template>
  <div>
    <div v-for="i in 3" :key="i">
      <div v-if="!render">{{i}}</div>
      <functional v-else :render="render" :val="i"></functional>
    </div>
  </div>
</template>
<script>
import { FunctionalRender, Exam } from '@vue/compiler-core'
export default {
  props: {
    render: {}
  },
  components: {
    FunctionalRender
  }
}
</script>

// 0x51FD;0x6570;0x5F0F;0x7EC4;0x4EF6;
export default {
  functional: true,
  render: (h, ctx) => {
    return ctx.props.render(h, ctx.props.val);
  }
}

```

可编辑表格

基于view使用jsx扩展成可编辑的表格

```

<template>
  <table height="200" :columns="innerColumn" :data="data"></table>
</template>
<script>
export default {
  data() {
    return {innerColumn: [], index: -1,}
  },
  methods: {
    editor(key) {
      this.index = key;
    },
    save() {
      this.index = -1;
    },
    change(row, column, value) {
      row[column['key']] = value
    }
  },
  props: {
    column: {},
    data: {}
  },
  mounted() {
    this.innerColumn = this.column.map(item => {
      if (item.edit) {
        item.render = (h, {row, index, column}) => {
          let flag = this.index == index
          return <div>
            {flag}

            <input value={row[column['key']]} type={flag ? 'text' : 'checkbox'} />
            {this.colName(row, column, value)}
          </div>
            <input type={row[column['key']]} />
          </div>
          {flag}

          <input type={flag ? 'text' : 'checkbox'} />
          删除
        </functional>
        <input type={flag ? 'text' : 'checkbox'} />
          保存
        </functional>
      }
      return item;
    })
  }
}
</script>

```