

# Vue全家桶原理剖析(三)

## Vue-router剖析

### Vue全家桶原理剖析(三)

0. 讲师介绍

1. 课前准备

2. 课堂主题

3. 课堂目标

4. 知识点

Vue-router入门代码

Vue.use

单页页面

hash模式

hishtory模式

代码才是王道

生命周期

5. 拓展

6. 本课总结

7. 作业

8. 问答

9. 下次课预告

# 0. 讲师介绍

---

## 1. 课前准备

---

1. Vue基础入门
2. vue-router基本使用

## 2. 课堂主题

---

1. Vue-router使用介绍
2. Vue插件机制
3. 前端单页面路由原理
4. 路由声明周期
5. 实现自己的vue-router
6. 后续扩展点

## 3.课堂目标

---

1. 掌握前端单页应用的原理
2. 掌握vue-router工作机制
3. 手写vue-router

## 4.知识点

---

新建工程

```
vue create krouter
```

## Vue-router入门代码

```
import Vue from 'vue'
import Router from 'vue-router'
import Home from './views/Home.vue'

Vue.use(Router)

export default new Router({
  base: process.env.BASE_URL,
```

```
routes: [  
  {  
    path: '/',  
    name: 'home',  
    component: Home  
  },  
  {  
    path: '/about',  
    name: 'about',  
    // route level code-splitting  
    // this generates a separate chunk  
(about.[hash].js) for this route  
    // which is lazy-loaded when the route is  
visited.  
    component: () => import(/*  
webpackChunkName: "about" */  
'./views/About.vue')  
  }  
]  
})
```

```
new Vue({  
  router,  
  render: h => h(App)  
}).$mount('#app')
```

# Vue.use

用户执行Vue.use的时候，实际执行的是模块的install方法，会把Vue的实例传递进去，比如咱们整个字符串

```
let Vue

class Router{

}

function install(_Vue){
  Vue = _Vue;
  Vue.mixin({
    beforeCreate() {
      const options = this.$options;
      console.log(options)
      Vue.prototype.$kkbrouter = '哈喽啊
小老弟'
    }
  })
}

export {install, Router}
```

# 单页页面

---

## hash模式

使用url#后面的锚点来区分组件，hash改变的时候，页面不会重新加载，只会出发onhashchange事件

## hishtory模式

hash的url略丑，使用mode:history，这种模式充分利用了html5 history interface 中新增的 pushState() 和 replaceState() 方法。这两个方法应用于浏览器记录栈，在当前已有的 back、forward、go 基础之上，它们提供了对历史记录修改的功能。只是当它们执行修改时，虽然改变了当前的 URL，但浏览器不会立即向后端发送请求。

## 代码才是王道

```
let Vue

class Router {
  static install(_Vue) {
    Vue = _Vue
    Vue.mixin({
```

```

        beforeCreate() {
            if(this.$options.router){
                console.log('>')
                // new Vue的时候传递的
                Vue.prototype.$kkbrouter =
                '哈喽啊 小老弟'
                Vue.prototype.$krouter =
                this.$options.router
                this.$options.router.init()
                //
                console.log(this.$options.router)
            }
            // console.log(options)
        }
    })
}
constructor(options) {
    this.$options = options
    this.routeMap = {}
    this.app = new Vue({
        data: {
            current: '/'
        }
    })
}

// 绑定事件

```

```
init() {
    this.bindEvents()
    this.createRouteMap(this.$options)
    this.initComponent(Vue)
}

bindEvents(){
    window.addEventListener('load',
this.onHashChange.bind(this), false)
    window.addEventListener('hashchange',
this.onHashChange.bind(this), false)

}

// 路由映射表
createRouteMap(options) {
    options.routes.forEach(item => {
        this.routeMap[item.path] =
item.component
    })
}

// 注册组件
initComponent(Vue) {
    Vue.component('router-link', {
        props: {
            to: String
        },
        render(h){
            console.log(this.to)
        }
    })
}
```



```

        return h('a', {
            attrs: {
                href: '#' + this.to
            }
        }, [
            this.$slots.default
        ])
    }
    // render
    // template: '<a :href="to">
<slot>123</slot></a>'
    })

    const _this = this
    Vue.component('router-view', {
        render(h) {
            var component =
_this.routeMap[_this.app.current]
            return h(component)
        }
    })
}

// 获取当前 hash 串
getHash() {
    return window.location.hash.slice(1) ||
'/'

```

```
}  
push(url){  
  window.location.hash = url  
  // 如果是  
  //  
window.history.pushState(null,null,'#/'+url)  
}  
// 设置当前路径  
onHashChange() {  
  this.app.current = this.getHash()  
}  
}  
  
export default Router
```

## 生命周期

```
beforeEnter(from,to,next){  
  console.log(`beforeEnterHome from  
${from} to ${to}`)  
  next()  
},
```

```
onHashChange(e) {  
    let {from, to} = this.getFrom(e)  
    let hash = this.getHash()  
    let router = this.routeMap[hash]  
  
    if(router.beforeEnter){  
        router.beforeEnter(from, to, ()=>{  
            this.app.current =  
this.getHash()  
        })  
    }else{  
        this.app.current = this.getHash()  
    }  
}
```

## 5. 拓展

---

1. 除了beforeCreate，还有有销毁
2. history模式
3. match
4. 生命周期

## 6. 本课总结

---

## 7. 作业

---

## 8. 问答

---

## 9. 下次课预告

---