

后盾网 人人做后盾

www.houdunwang.com

router

后盾网 2011-2017 v2.0

vue-router用于控制单页面SPA应用中显示不同页面的js库。

手册

<https://router.vuejs.org/zh-cn/installation.html>

直接下载 / CDN

<https://unpkg.com/vue-router/dist/vue-router.js>

NPM

```
npm install vue-router
```

view-router

router-link用于设置跳转的地址，router-view用于显示组件的容器。

```
<div id="app">
  <router-link to="/hdphp">hdphp</router-link>
  <router-link to="/hdcms">hdcms</router-link>
  <!-- 路由匹配到的组件将渲染在这里 -->
  <router-view></router-view>
</div>
<script>
  //组件
  hdphp = {template: '<h1>我是 hdphp</h1>'};
  hdcms = {template: '<h1>我是 hdcms</h1>'};
  //路由规则名称必须为routes这是vue-router的属性
  routes = [
    {path: '/hdphp', component: hdphp},
    {path: '/hdcms', component: hdcms}
  ];
  //创建路由实例,实例名称必须为router
  router = new VueRouter({routes});
  new Vue({
    el: '#app',
    router
  })
</script>
```

定义路由

当 `<router-link>` 对应的路由匹配成功, 将自动设置 `class` 属性值 `.router-link-active`, 需要为 `router-link` 添加上 `exact` 属性指当路由完全匹配时添加 `class`。

```
<style>
  a.router-link-active {
    background: red;
    color: #fff;
  }
</style>
<div id="app">
  <router-link to="/hd" exact>首页</router-link>
  <router-link to="/hd/login">登录</router-link>
  <router-view></router-view>
</div>
<script>
  home = {template: '<h1>后盾官网</h1>'}
  login= {template: '<h1>学员登录</h1>'}
  routes = [
    {path: '/hd', component: home},
    {path: '/hd/login', component: login}
  ];
  router = new VueRouter({routes});
  new Vue({
    el: '#app',
    router
  })
</script>
```

当前链接添加样式

有时候想同时（同级）展示多个视图，而不是嵌套展示，例如创建一个布局，有 sidebar（侧导航）和 main（主内容）两个视图，这个时候命名视图就派上用场了。你可以在界面中拥有多个单独命名的视图。如果 router-view 没有设置名字，那么默认为 default。

```
<div id="app">
  <router-view name="header"></router-view>
  <router-view></router-view>
  <router-view name="footer"></router-view>
</div>
<script>
  hdHeader = {template: '<div>顶部</div>'};
  hdFooter = {template: '<div>底部</div>'};
  lists = {template: '<div>列表</div>'};
  //定义路由
  const routes = [{
    path: '/', components: {
      default: lists,
      header: hdHeader,
      footer: hdFooter
    }
  }];
  //创建路由实例
  router = new VueRouter({routes});
  app = new Vue({
    el: '#app',
    router
  });
</script>
```

命名视图



组件要根据不同的参数显示不同的数据，这需要路由可以传参。

```
<div id="app">
  <router-link :to="{name: 'news', params: {id: 1}}">新闻1</router-link>
  <router-link :to="{name: 'news', params: {id: 2}}">新闻2</router-link>
  <!-- 路由匹配到的组件将渲染在这里 -->
  <router-view></router-view>
</div>
<script>
  // 组件
  const news = {template: '<div>新闻 id:{{this.$route.params.id}}</div>'};
  const routes = [
    {path: '/news/:id', component: news, name: 'news'},
  ];
  // 创建路由实例
  const router = new VueRouter({routes});
  const app = new Vue({
    el: '#app',
    router
  });
</script>
```

路由参数

当没有路由匹配时使用默认路由指令的组件，可用于设置首页或404页面。只需要设置路由path值为'*'即可定义默认路由。

```
<div id="app">
  <router-link to="/home">首页</router-link>
  <router-link to="/content/1">播放页</router-link>
  <router-view></router-view>
</div>
<script>
  const home = {template: '<div>首页</div>'};
  const content = {template: '<div>{{this.$route.params.id}}</div>'};
  const def = {template: '<div>默认组件</div>'};
  const routes = [
    {path: '/home', component: home},
    {path: '/content/:id', component: content},
    {path: '*', component: def}
  ];
  //创建路由实例
  const router = new VueRouter({routes});
  const app = new Vue({
    el: '#app',
    router
  });
</script>
```

默认路由