

vuex-router-sync 理解

作者: [Rabbitzzc](#)

原文链接: <https://ld246.com/article/1597060816093>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

vuex-router-sync 基本原理理解

先说结论：把 `vue-router` 当前的 `$route` 同步为 `vuex` 状态的一部分。换句话说，你不需要再 `Vuex` `import route`，而可以直接使用 `$route`

使用方法

如果你的项目中，在 `Vuex` 中不需要获取 `route`，则完全没必要引用 `vuex-router-sync`，但是如果尝试在 `vuex` 方法中使用 `route` 对象，则需要引入。

比如需要在组件中显示一条消息。首页显示'这里是聊天室',其他页面则显示'感谢访问':

```
const Top = {
  template: '<div>{{message}}</div>',
  computed: {
    message() {
      return this.$store.getters.getMessage;
    }
  },
};
const Bar = {
  template: '<div>{{message}}</div>',
  computed: {
    message() {
      return this.$store.getters.getMessage;
    }
  }
};

const routes = [{
  path: '/top',
  component: Top,
  name: 'top'
},
{
  path: '/bar',
  component: Bar,
  name: 'bar'
}
];

const router = new VueRouter({
  routes
});

const store = new Vuex.Store({
  state: {
    username: 'Rabbitzzc',
    phrases: ['欢迎进入聊天室HacPai', '感谢访问'],
  },
  getters: {
    getMessage(state) {
      return state.route.name === 'top' ?
```

```
    `${state.phrases[0]}, ${state.username}` :  
    `${state.phrases[1]}, ${state.username}`;  
  },  
},  
});
```

```
sync(store, router);
```

```
const app = new Vue({  
  router,  
  store,  
}).$mount('#app');
```

原理

源码地址: <https://github.com/vuejs/vuex-router-sync/blob/master/README.zh-cn.md>

该库是在 `store` 上增加了一个名为 `route` 的模块, 用于表示当前路由的状态

```
store.state.route.path // current path (字符串类型)  
store.state.route.params // current params (对象类型)  
store.state.route.query // current query (对象类型)
```

当被导航到一个新路由时, `store` 的状态会被更新。

`store.state.route` 是不可变更的, 因为该值取自 URL, 是真实的来源。你不应该通过修改该值去触发浏览器的导航行为。取而代之的是调用 `$router.push()` 或者 `$router.go()`。另外, 你可以通过 `$router.push({ query: {...}})` 来更新当前路径的查询字符串。

换句话说就是将 `route` 模块复制到了 `store` 上面, 并且是一个可读属性。

整体逻辑

写了一个demo: <https://codesandbox.io/s/relaxed-currying-sq37c?file=/src/lib/sync.js>

demo中将一些注释已经备注了, 大概画了一下流程图:

利用registerModule向Vuex动态注册Module（默认名称为 route），设置state与ROUTE_CHANGED mutations

利用store.watch监听store内部store.state变化

是

isTimeTraveling = true

路由跳转
router.push/router.replace

否，为外部触发，
比如外部调用push等

触发 mutation
store.commit(moduleName +
'/ROUTE_CHANGED')

利用 router.afterEach
监听路由变化

是

判断Boolean: isTimeTraveling

false