

Vue Router

Routing in a Vue App

Client-Side vs. Server-Side Routing

Routing on the server side means the server sending a response based on the URL path that the user is visiting.

In a [Single-Page Application](#) (SPA), however, the client-side JavaScript can intercept the navigation, dynamically fetch new data, and update the current page without full page reloads.

Simple Routing

Remember "Dynamic
Components"?

Simple Routing

Remember "Dynamic Components"?

```
1 <script>
2 import Home from './Home.vue'
3 import About from './About.vue'
4 import NotFound from './NotFound.vue'
5
6 const routes = {
7   '/': Home,
8   '/about': About
9 }
10
11 export default {
12   data() {
13     return {
14       currentPath: window.location.hash
15     }
16   },
17   computed: {
18     currentView() {
19       return routes[this.currentPath.slice(1) || '/'] || NotFound
20     }
21   },
22   mounted() {
23     window.addEventListener('hashchange', () => {
24       this.currentPath = window.location.hash
25     })
26   }
27 }
28 </script>
```

Simple Routing

Remember "Dynamic Components"?

```
1 <template>
2   <a href="#/">Home</a> |
3   <a href="#/about">About</a> |
4   <a href="#/non-existent-path">Broken Link</a>
5   <component :is="currentView" />
6 </template>
```

```
1 <script>
2 import Home from './Home.vue'
3 import About from './About.vue'
4 import NotFound from './NotFound.vue'
5
6 const routes = {
7   '/': Home,
8   '/about': About
9 }
10
11 export default {
12   data() {
13     return {
14       currentPath: window.location.hash
15     }
16   },
17   computed: {
18     currentView() {
19       return routes[this.currentPath.slice(1) || '/'] || NotFound
20     }
21   },
22   mounted() {
23     window.addEventListener('hashchange', () => {
24       this.currentPath = window.location.hash
25     })
26   }
27 }
28 </script>
```

Simple Routing

Remember "Dynamic Components"?

```
1 <template>
2   <a href="#/">Home</a> |
3   <a href="#/about">About</a> |
4   <a href="#/non-existent-path">Broken Link</a>
5   <component :is="currentView" />
6 </template>
```

```
1 <script>
2 import Home from './Home.vue'
3 import About from './About.vue'
4 import NotFound from './NotFound.vue'
5
6 const routes = {
7   '/': Home,
8   '/about': About
9 }
10
11 export default {
12   data() {
13     return {
14       currentPath: window.location.hash
15     }
16   },
17   computed: {
18     currentView() {
19       return routes[this.currentPath.slice(1) || '/'] || NotFound
20     }
21   },
22   mounted() {
23     window.addEventListener('hashchange', () => {
24       this.currentPath = window.location.hash
25     })
26   }
27 }
28 </script>
```

Simple Routing

Remember "Dynamic Components"?

```
1 <template>
2   <a href="#/">Home</a> |
3   <a href="#/about">About</a> |
4   <a href="#/non-existent-path">Broken Link</a>
5   <component :is="currentView" />
6 </template>
```

```
1 <script>
2   import Home from './Home.vue'
3   import About from './About.vue'
4   import NotFound from './NotFound.vue'
5
6   const routes = {
7     '/': Home,
8     '/about': About
9   }
10
11   export default {
12     data() {
13       return {
14         currentPath: window.location.hash
15       }
16     },
17     computed: {
18       currentView() {
19         return routes[this.currentPath.slice(1) || '/'] || NotFound
20       }
21     },
22     mounted() {
23       window.addEventListener('hashchange', () => {
24         this.currentPath = window.location.hash
25       })
26     }
27   }
28 </script>
```

Vue Router

Creating a Single-page Application with Vue + Vue Router feels natural, but it needs to be imported separate. Luckily it can't be easier with create-vue.

Vue Router - Exercise

In 08_vue_router you will find an example app.
Please run (inside the folder):



```
1 npm install  
2 npm run dev
```

Vue Router - Exercise

You will see the following:



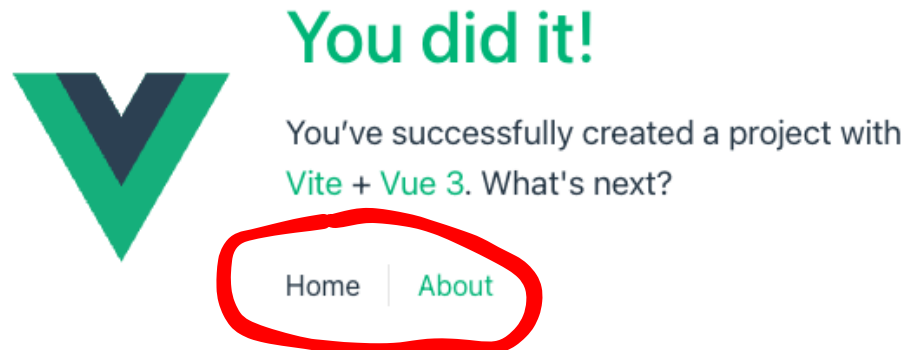
You did it!

You've successfully created a project with
Vite + Vue 3. What's next?

[Home](#) | [About](#)

Vue Router - Exercise

You will see the following:



Vue Router - Exercise

Since you have quite an experience, please have a look what has changed or what "is new". Can you explain where the "Vue Router" acts?

Vue Router - Solution

App.vue



```
1 import { RouterLink, RouterView } from 'vue-router'
```



```
1 <nav>
2   <RouterLink to="/">Home</RouterLink>
3   <RouterLink to="/about">About</RouterLink>
4 </nav>
```

Vue Router - Solution

router.ts

```
1 import { createRouter, createWebHistory } from 'vue-router'
2 import HomeView from '../views/HomeView.vue'
3
4 const router = createRouter({
5   history: createWebHistory(import.meta.env.BASE_URL),
6   routes: [
7     {
8       path: '/',
9       name: 'home',
10      component: HomeView
11    },
12    {
13      path: '/about',
14      name: 'about',
15      // route level code-splitting
16      // this generates a separate chunk (About.[hash].js) for this route
17      // which is lazy-loaded when the route is visited.
18      component: () => import('../views/AboutView.vue')
19    }
20  ]
21 })
22 export default router
```

Vue Router - Exercise

Please try to add a new "Route". Hint: 3 Files need to be changed.

Vue Router - Exercise

What is the difference between:



```
1 component: HomeView
```

and:



```
1 component: () => import('../views/AboutView.vue')
```


Dynamic Route Matching with Params

Very often we will need to map routes with the given pattern to the same component.

```
1 const User = {  
2   template: '<div>User</div>',  
3 }  
4  
5 // these are passed to `createRouter`  
6 const routes = [  
7   // dynamic segments start with a colon  
8   { path: '/users/:id', component: User },  
9 ]
```

Now URLs like **/users/johnny** and **/users/jolyne** will both map to the same route.

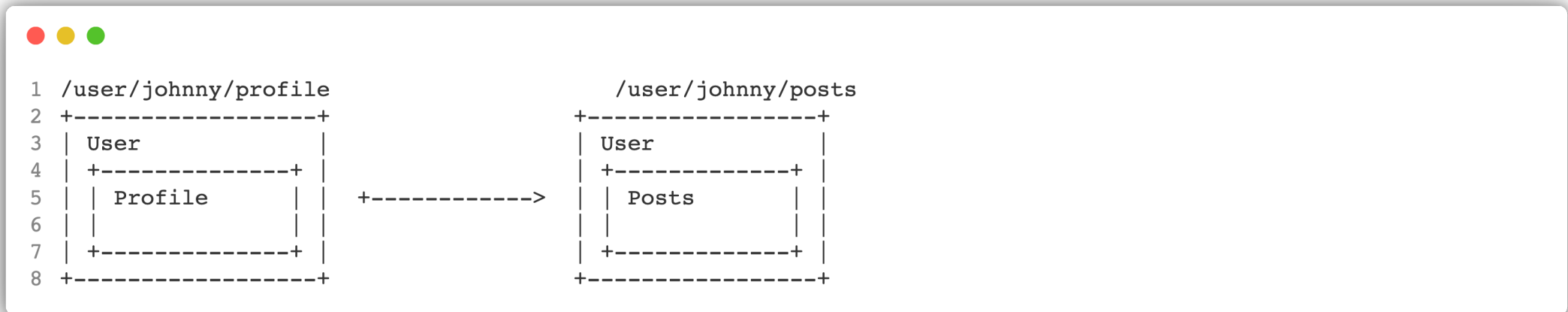
Routes' Matching Syntax

The matching syntax is quite mighty, and one thing is regex (others are omitted for simplicity)

```
1 const routes = [  
2   // /:orderId -> matches only numbers  
3   { path: '/:orderId(\\d+)' },  
4   // /:productName -> matches anything else  
5   { path: '/:productName' },  
6 ]
```

Nested Routes

Some application's UIs are composed of components that are nested multiple levels deep.



Nested Routes

With Vue Router, you can express this relationship using nested route configurations.



```
1 <div id="app">
2   <router-view></router-view>
3 </div>
```



```
1 const User = {
2   template: '<div>User {{ $route.params.id }}</div>',
3 }
4
5 // these are passed to `createRouter`
6 const routes = [{ path: '/user/:id', component: User }]
```

Nested Routes

With Vue Router, you can express this relationship using nested route configurations.

```
1 const User = {  
2   template: `  
3     <div class="user">  
4       <h2>User {{ $route.params.id }}</h2>  
5       <router-view></router-view>  
6     </div>  
7   `,  
8 }
```

Nested Routes

With Vue Router, you can express this relationship using nested route configurations.

```
1  const routes = [  
2    {  
3      path: '/user/:id',  
4      component: User,  
5      children: [  
6        {  
7          // UserProfile will be rendered inside User's <router-view>  
8          // when /user/:id/profile is matched  
9          path: 'profile',  
10         component: UserProfile,  
11       },  
12       {  
13         // UserPosts will be rendered inside User's <router-view>  
14         // when /user/:id/posts is matched  
15         path: 'posts',  
16         component: UserPosts,  
17       },  
18     ],  
19   },  
20 ]
```

Different History modes - Hash

The hash history mode is created with `createWebHashHistory()`:

```
1 import { createRouter, createWebHashHistory } from 'vue-router'
2
3 const router = createRouter({
4   history: createWebHashHistory(),
5   routes: [
6     //...
7   ],
8 })
```

Different History modes - HTML5

The HTML5 mode is created with `createWebHistory()` and is the **recommended mode**:

```
1 import { createRouter, createWebHistory } from 'vue-router'
2
3 const router = createRouter({
4   history: createWebHistory(),
5   routes: [
6     //...
7   ],
8 })
```


Different History modes - Memory Mode

The memory history mode doesn't assume a browser environment and therefore doesn't interact with the URL nor automatically triggers the initial navigation.

```
1 import { createRouter, createMemoryHistory } from 'vue-router'
2
3 const router = createRouter({
4   history: createMemoryHistory(),
5   routes: [
6     //...
7   ],
8 })
```

End



End

That was all for this chapter
