



# ROUTER TRONG VUE.JS



## III NỘI DUNG CHÍNH

I

GIỚI THIỆU ROUTER

II

VUE ROUTER BASIC

III

CÁC LOẠI SLOT

IV

SCOPED SLOT

## 1. Vue Router

### Tại sao sử dụng Vue Router:

- Thế mạnh của Vue là ứng dụng tạo trang đơn (Single Page). Các trang web có tính tương tác cao không làm mới khi bạn thay đổi từ trang này sang trang khác. Nếu trang web của bạn có nhiều trang (hoặc “lượt xem”) và bạn đang sử dụng Vue, thì đó là lý do tại sao bạn cần Vue Router.

## 2. Vue Router Basic

### Xây dựng Router đơn giản

- Ta thiết lập trang cli như sau:

```
$ npm install -g @vue/cli  
or  
$ yarn global add @vue/cli
```

- Sau đó tạo ứng dụng cơ bản tên my-router-project

```
$ vue create my-router-project
```

# 1. Vue Router

## Xây dựng Router đơn giản

- Chọn “Vue 3 Preview”

```
? Please pick a preset:  
  Default ([Vue 2] babel, eslint)  
➤ Default (Vue 3 Preview) ([Vue 3] babel, eslint)  
  Manually select features
```

- Chọn ứng dụng Vue 3 và đi đến đường dẫn, sau đó run server

```
$ cd my-router-project  
  
then  
  
$ npm run serve
```

Sau đó truy cập trang  
[http://localhost:8080](http://localhost:8080)

## 2. Vue Router Basic

- Để cài đặt Router vào project Vue 3 cơ sở (như chúng ta đã tạo ở trên) hoặc vào một project hiện có đã sử dụng Vue 3, chúng ta sẽ làm theo các bước dưới đây:

### 1. Install Vue Router từ command Line

Vào project vừa tạo chạy lệnh

```
$ npm i vue-router@next
```

### 2. Thêm đường dẫn Routing & cấu hình `/src/router/index.js` như sau:

➤ Tạo thư mục tên router trong src

➤ Tạo file tên index.js trong thư mục router với đoạn code như sau

## 2. Vue Router Basic

```
import { createWebHistory, createRouter } from "vue-router";
import Home from "@views/Home.vue";
import About from "@views/About.vue";

const routes = [
  {
    path: "/",
    name: "Home",
    component: Home,
  },
  {
    path: "/about",
    name: "About",
    component: About,
  },
];

const router = createRouter({
  history: createWebHistory(),
  routes,
});

export default router;
```

Chúng ta **export router**  
cuối vì chúng ta cần  
**import** vào trong main.js

## 2. Vue Router Basic

- **Path:** đường dẫn URL nơi có thể tìm thấy route
- **Name:** Tên tùy chọn để sử dụng khi chúng ta liên kết route.
- **Component:** Thành phần nào sẽ được load khi route được gọi

## 3. Import Router và sử dụng Vue Router

Chỉnh sửa file main.js như sau:

```
import { createApp } from 'vue'  
import App from './App.vue'  
import router from './router' // <---  
  
createApp(App).use(router).mount('#app')
```

Import Router vào  
main.js

Sử dụng “**use**” khi  
tạo ra ứng dụng



## 2. Vue Router Basic

### 4. Sử dụng `<router-view>` Và `<router-link>`

`<router-view />`- Khi **route** được điều hướng đến (navigate to) trong trình duyệt, đây là nơi component được hiển thị. Ví dụ, trong đoạn mã của chúng ta /sẽ hiển thị thành phần Home ở nơi chúng ta liệt kê `<router-view />`.

`<router-link>`- Đây là directive (chỉ thị) chúng ta sử dụng để tạo liên kết giữa các trang thành phần khác nhau, thay vì sử dụng `<a href>`.

Tạo một App.vue như sau:

## 1. Vue Router Basic

**/src/App.vue**

```
<template>
  <div id="nav">
    <router-link to="/">Home</router-link> |
    <router-link to="/about">About</router-link>
  </div>
  <router-view />
</template>
```

## 2. Vue Router Basic

- Thay vì chúng ta tạo trong đường dẫn **/components/**. Chúng ta sẽ tạo trong đường dẫn **View**

**/src/views/Home.vue**

```
<template>  
  <h1>Home Page</h1>  
</template>
```

**/src/views/About.vue**

```
<template>  
  <h1>About Page</h1>  
</template>
```



### 3. Install từ CLI

- Phần lớn cấu hình và thiết lập ở trên thực sự có thể nhờ Vue 3 làm từ CLI, nhưng chúng ta nên biết cách tự thực hiện. Tạo một project tên my-router-project như sau:

```
$ vue create my-router-project
```

- Sau đó chọn “Manually select features”

```
? Please pick a preset:  
  Default ([Vue 2] babel, eslint)  
  Default (Vue 3 Preview) ([Vue 3] babel, eslint)  
> Manually select features
```

### 3. Install từ CLI

- **Chọn Router**

```
? Check the features needed for your project:
```

- ☒ Choose Vue version
- ☒ Babel
- ☐ TypeScript
- ☐ Progressive Web App (PWA) Support
- ☒ Router
- ☐ Vuex
- ☐ CSS Pre-processors
- ☒ Linter / Formatter
- ☐ Unit Testing
- ☐ E2E Testing

### 3. Install từ CLI

- Phiên bản mới nhất của Vue Router sẽ được cài đặt và nó sẽ tạo trước hầu hết các cấu hình giống như chúng ta đã tạo ở trên.
- **Name Router**

/src/App.vue

```
<router-link :to="{ name: 'Home' }">Home</router-link> |  
<router-link :to="{ name: 'About' }">About</router-link>
```

- Chúng ta không sử dụng đường dẫn. Điều này hữu ích nếu đường dẫn URL của chúng ta trong tương lai thay đổi. Chúng ta sẽ không phải thay đổi tất cả router-links nếu đường dẫn thay đổi, vì chúng đang tham chiếu bằng cách sử dụng tên (**name**).

### 3. Install từ CLI

- **Dynamic Segments**
- Nếu chúng tôi muốn tạo URL đường dẫn có một biến, như `/user/gregg` hoặc `event/5` trong đó 'gregg' hoặc '5' có thể là bất kỳ tên nào hoặc bất kỳ số nào, chúng ta cần Dynamic Segments. Điều này được thực hiện trong Vue 2.
- Trong bộ router, chúng ta chỉ định route với `:` biểu thị biến.

`/src/router/index.js`

```
{  
  path: "/user/:name", // <-- notice the colon  
  name: "User",  
  component: User,  
},
```

### 3. Install từ CLI

- Để truy cập Dynamic segment trong view, chúng ta viết như sau:

**/src/views/User.vue**

```
<template>  
  <h1>The user is {{ $route.params.name }}</h1>  
</template>
```



**The user is Gregg**



### 3. Install từ CLI

- Cũng giống như Vue 2, Chúng ta có thể chọn dynamic segment chuyển vào component như là prop, nhưng chỉ định props = true.

**/src/router/index.js**

```
{  
  path: "/user/:name", // <-- notice the colon  
  name: "User",  
  component: User,  
  props: true,  
},
```

### 3. Install từ CLI

- Bên trong component chúng ta có

**/src/views/User.vue**

```
<template>
  <h1>The user is {{ name }}</h1>
</template>
<script>
export default {
  props: ["name"],
};
</script>
```

Lưu ý: Phải chỉ định  
component đang  
mong đợi props

**The user is Gregg**

### 3. Install từ CLI

- Chúng ta làm ứng dụng single page, nhận thấy rằng nếu chúng ta truy cập vào bất kỳ URL nào mà không có ánh xạ đến file route, chúng ta sẽ nhận được một trang chưa hoàn chỉnh. Nó tải lên từ App.vue nhưng không có thành phần nào. Chúng ta sẽ cần cho route biết phải làm gì khi nó không khớp với bất kỳ route nào và cách chúng ta sẽ thực hiện trong Vue 3.
- Chúng ta cần tạo một route ở cuối danh sách routing như sau:

**/src/router/index.js**

```
{  
  path: "/*",  
  component: NotFound,  
},
```

### 3. Install từ CLI

- **Lưu ý:** Đừng quên thêm NotFound vào đầu component

**/src/views/NotFound.vue**

```
<template>
  <h1>Oops, it looks like the page you're looking for doesn't exist.</h1>
</template>
```



The  
end