



TRƯỜNG ĐẠI HỌC LẠC HỒNG

LẬP TRÌNH WEB NÂNG CAO

BÀI 05 XÂY DỰNG ỨNG DỤNG

Giáo viên: Trần Thanh Phương

Email: thanhphuong@lhu.edu.vn

Nội dung bài học

- 01 Tạo Project Front-End sử dụng Vue Template
- 03 Khai báo thư viện axios để gọi api
- 04 Khai báo thư viện vue-router để định tuyến trang web
- 05 Tạo trang Login để chứng thực người dùng
- 06 Tạo trang hiển thị thông tin



TẠO PROJECT FRONT-END

1. Vào Command Prompt và chuyển đến thư mục chứa Project cần tạo
Nhập lệnh như hình bên dưới

```
C:\WINDOWS\system32\cmd.exe

D:\>vue create fe_demo_
```

```
C:\WINDOWS\system32\cmd.exe

Vue CLI v5.0.8
? Please pick a preset: (Use arrow keys)
> Default ([Vue 3] babel, eslint)
  Default ([Vue 2] babel, eslint)
  Manually select features
```

Chọn version của Vue

```
C:\WINDOWS\system32\cmd.exe

Vue CLI v5.0.8
✔ Creating project in D:\fe_demo.
✔ Installing CLI plugins. This might take a while...

yarn install v1.22.22
info No lockfile found.
[1/4] Resolving packages...
✔ @babel/helper-plugin-utils@7.24.7
```

Quá trình tạo dự án và cài đặt các thư viện cần thiết



TẠO PROJECT FRONT-END

2. Thêm thư viện **Vuetify** (mục đích là sử dụng được các component của Vue chẳng hạn như button, textbox, ...)

```
C:\WINDOWS\system32\cmd.exe

D:\fe_demo>vue add vuetify
```

```
C:\WINDOWS\system32\cmd.exe

rechoir@0.6.2
shelljs@0.8.5
vue-cli-plugin-vuetify@2.5.8
Done in 6.60s.
[+] Successfully installed plugin: vue-cli-plugin-vuetify

? Choose a preset:
  Vuetify 2 - Configure Vue CLI (advanced)
  Vuetify 2 - Vue CLI (recommended)
  Vuetify 2 - Prototype (rapid development)
  Vuetify 3 - Vite (preview)
> Vuetify 3 - Vue CLI (preview)
```

Chọn version của Vue



TẠO PROJECT FRONT-END

3. Thêm thư viện **router** (chức năng dùng để tạo liên kết điều hướng trang web)

```
C:\WINDOWS\system32\cmd.exe

[3/4] Linking dependencies...
[4/4] Building fresh packages...
success Saved lockfile.
Done in 6.11s.
  Running completion hooks...

  Successfully invoked generator for plugin: vue-cli-plugin-vuetify
vuetify  Discord community: https://community.vuetifyjs.com
vuetify  Github: https://github.com/vuetifyjs/vuetify
vuetify  Support Vuetify: https://github.com/sponsors/johnleider

D:\fe_demo>yarn add vue-router
```



TẠO PROJECT FRONT-END

4. Thêm thư viện **axios** (dùng để gọi api)

```
C:\WINDOWS\system32\cmd.exe

D:\fe_demo>yarn add axios_
```

```
C:\WINDOWS\system32\cmd.exe

[4/4] Building fresh packages...
success Saved lockfile.
success Saved 6 new dependencies.
info Direct dependencies
└─ axios@1.7.7
info All dependencies
├─ asynckit@0.4.0
├─ axios@1.7.7
├─ combined-stream@1.0.8
├─ delayed-stream@1.0.0
├─ form-data@4.0.0
└─ proxy-from-env@1.1.0
Done in 5.25s.

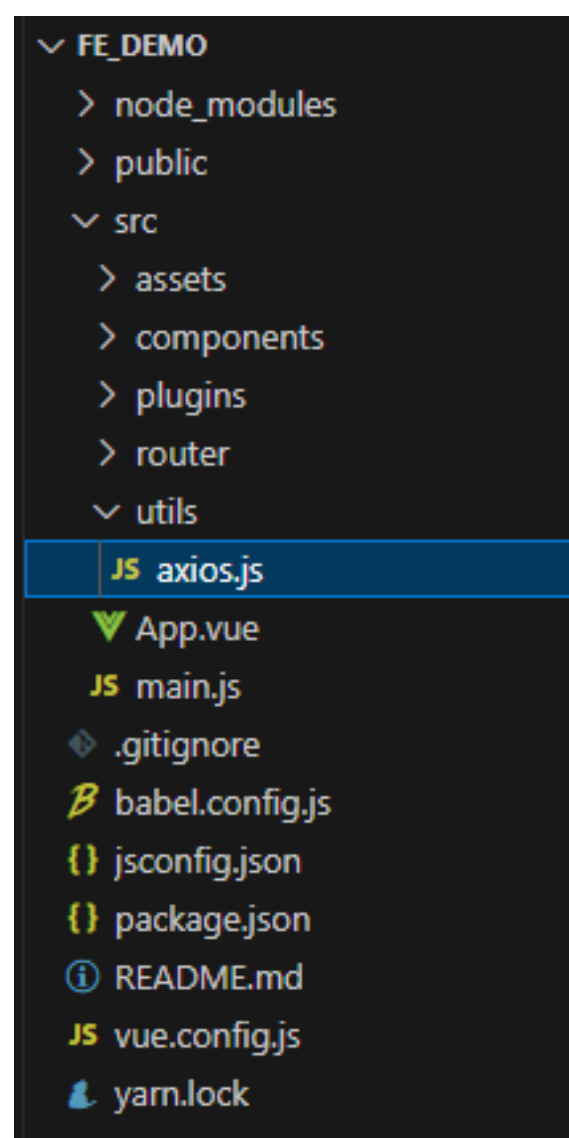
D:\fe_demo>
```

Cài đặt thành công



KHAI BÁO THƯ VIỆN AXIOS ĐỂ GỌI API

1. Tạo thư mục **utils** trong thư mục **src**
2. Tạo file **axios.js** trong thư mục **utils** vừa tạo



Nếu không có chứng thực thì chuyển đến trang login

```
src > utils > JS axios.js > ...
1  import axios from "axios";
2  ⚡
3  let auth = localStorage.getItem('auth');
4  if (auth == '') location = 'http://localhost:8080/login';
5
6  const instance = axios.create({
7    baseUrl: 'http://localhost:5178',
8    headers: { Authorization: 'Bearer ' + auth }
9  });
10
11 instance.interceptors.request.use(function (config) {
12   // Do something before request is sent
13   return config;
14 }, function (error) {
15   // Do something with request error
16   return Promise.reject(error);
17 });
18
19 // Add a response interceptor
20 instance.interceptors.response.use(function (response) {
21   // Any status code that lie within the range of 2xx cause this function to trigger
22   // Do something with response data
23   return response;
24 }, function (error) {
25   // Any status codes that falls outside the range of 2xx cause this function to trigger
26   // Do something with response error
27   return Promise.reject(error);
28 });
29
30 export default instance
```

Chuỗi token sau khi đăng nhập thành công được lưu trữ trong **localStorage** hoặc **Cookie** với tên là **auth** (tên này là do người lập trình tự đặt tùy ý)

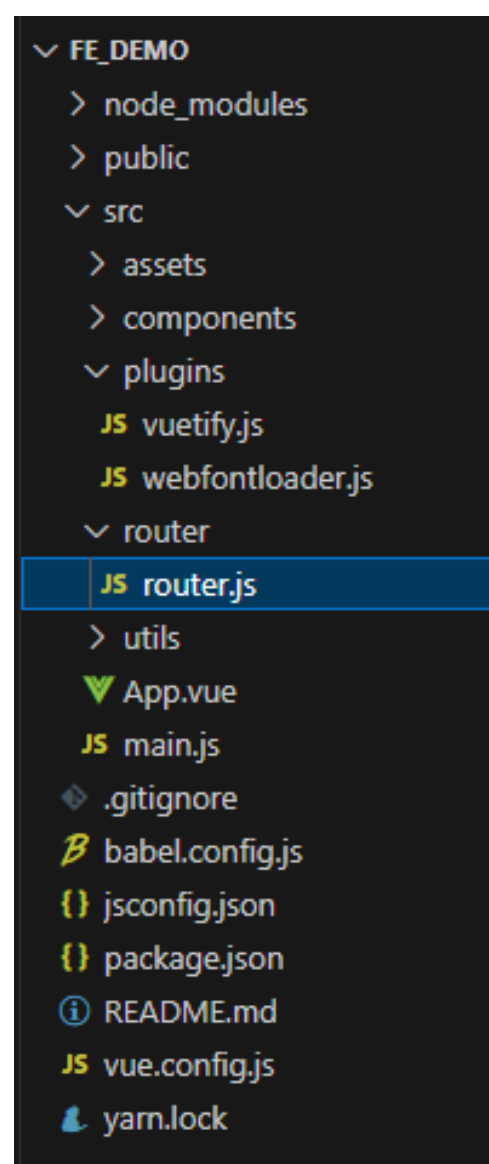
Địa chỉ của api domain

Chèn chuỗi **token** và phần **Header** khi gọi api



KHAI BÁO THƯ VIỆN VUE-ROUTER ĐỂ ĐỊNH TUYẾN TRANG WEB

1. Tạo thư mục **router** trong thư mục **src**
2. Tạo file **router.js** trong thư mục **router** vừa tạo



```
src > router > JS router.js > [0] default
1  import { createRouter, createWebHistory } from 'vue-router';
2  import LoginComponent from '../components/login.vue';
3  import ChuDeComponent from '../components/chu-de.vue';
4  import HomeComponent from '../components/home.vue';
5
6  const routes = [
7    {
8      path: '/home',
9      name: 'HomeComponent',
10     component: HomeComponent,
11     meta: { requiresAuth: true }
12   },
13   {
14     path: '/login',
15     name: 'LoginComponent',
16     component: LoginComponent
17   },
18   {
19     path: '/chu-de',
20     name: 'ChuDeComponent',
21     component: ChuDeComponent,
22     meta: { requiresAuth: true }
23   },
24   {
25     path: '/*',
26     redirect: '/login',
27   }
28 ];
29
30 const router = createRouter({
31   history: createWebHistory(),
32   routes,
33 });
34
35 // Middleware kiểm tra đăng nhập
36 router.beforeEach((to, from, next) => {
37   const isAuthenticated = localStorage.getItem('auth'); // Hoặc kiểm tra token từ store
38
39   if (to.meta.requiresAuth && !isAuthenticated) {
40     // Nếu cần đăng nhập mà chưa đăng nhập, điều hướng về trang login
41     next('/login');
42   } else {
43     // Nếu đã đăng nhập hoặc không cần bảo mật, cho phép điều hướng
44     next();
45   }
46 });
47
48 export default router;
```

Khai báo thư viện

Khai báo các trang

Khai báo menu

Có yêu cầu chứng thực khi gọi trang này

Người dùng nhập đường dẫn không tồn tại sẽ tự động điều hướng đến trang login

Trước khi điều hướng kiểm tra xem trang đang gọi có yêu cầu chứng thực hay không



KHAI BÁO THƯ VIỆN VUE-ROUTER ĐỂ ĐỊNH TUYẾN TRANG WEB

3. Mở file main.js và bổ sung các dòng lệnh như hình bên dưới

```
src > JS main.js
1  import { createApp } from 'vue'
2  import App from './App.vue'
3  import vuetify from './plugins/vuetify'
4  import { loadFonts } from './plugins/webfontloader'
5  import router from './router/router'
6
7  loadFonts()
8
9  createApp(App)
10   .use(vuetify)
11   .use(router)
12   .mount('#app')
13
```

Tạo trang login chứng thực người dùng

1. Tạo thủ tục



```
create proc [dbo].[spAdmin_UserLogin]
    @UserName varchar(30),
    @PassWord varchar(50)
as
begin try
    if not exists (
        select *
        from tbl_Account
        where UserName = @UserName and PWDCOMPARE (@PassWord, UserPass) = 1 and UserStatus = 1
    )
    begin
        raiserror (N'Tên đăng nhập hoặc mật khẩu không chính xác.', 16, 1)
        return
    end

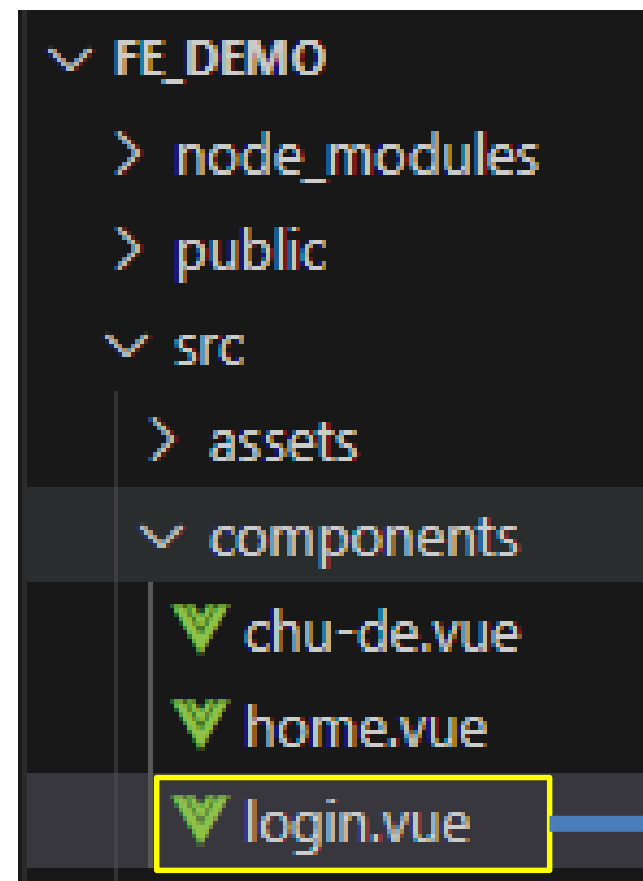
    update tbl_Account
    set LastLogin = getdate()
    where UserName = @UserName

    select UserName, FullName, AllowInsert, AllowUpdate, AllowDelete,
           UserRight, UserStatus, LastLogin, UpdateTime
    from tbl_Account
    where UserName = @UserName
end try
begin catch
    declare @err nvarchar(1000) = ERROR_MESSAGE()
    raiserror (@err, 16, 1)
end catch
```

2. Tạo trang login



Tạo file login.vue trong thư mục components của dự án



Đăng nhập thành công điều hướng đến trang home

```
<template>
  <v-sheet class="mx-auto mt-4" width="400">
    <div class="text-center mb-2"><b>Đăng nhập hệ thống</b></div>
    <v-form>
      <v-text-field v-model="userName" label="Tên đăng nhập"></v-text-field>
      <v-text-field v-model="userPass" label="Mật khẩu"></v-text-field>
      <v-btn class="mt-2" block @click="userLogin()" color="primary" elevation="0">Đăng nhập</v-btn>
    </v-form>
    <div v-if="loginError != ''" class="text-center mt-4 mb-2 error--text">{{ loginError }}</div>
  </v-sheet>
</template>

<script>
import axios from '../utils/axios'

export default {
  name: 'LoginComponent',
  data: () => ({
    userName: '',
    userPass: '',
    loginError: ''
  }),
  methods: {
    userLogin() {
      axios.post('/user/login', {
        UserName: this.userName,
        Password: this.userPass
      }).then(function (response) {
        localStorage.setItem('auth', response.data.token)
        location = "/home"
      }).catch((error) => {
        this.loginError = error.response.data.message;
      });
    }
  }
}
```

Sự kiện click vào nút đăng nhập

Import thư viện để gọi api

Giao diện trang login

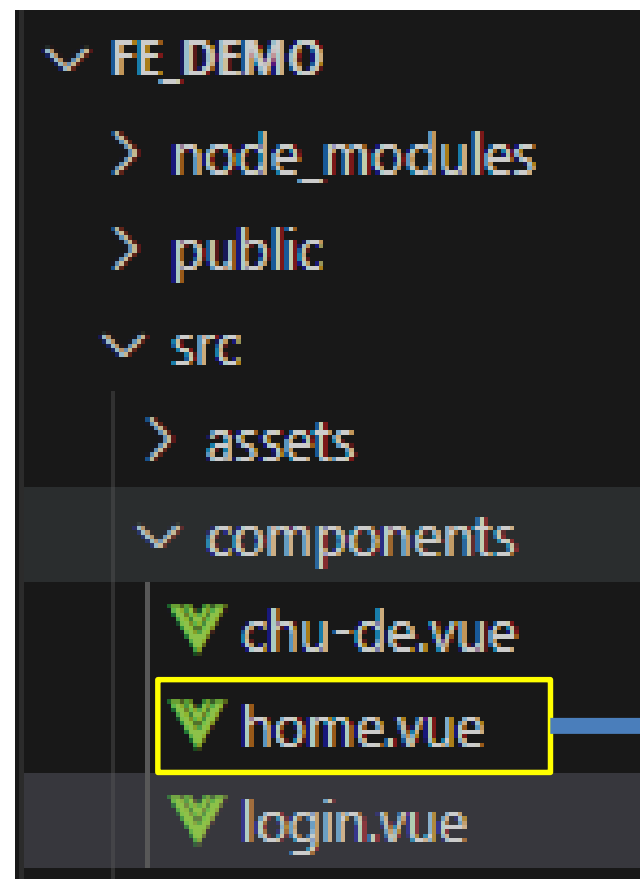
Nếu đăng nhập thành công thì lưu chuỗi token vào localStorage với tên là auth

Thông báo lỗi khi đăng nhập không thành công

3. Tạo trang home



Tạo file home.vue trong thư mục components của dự án



```
src > components > ▼ home.vue > {} script
1   <template>
2   |   <div>Welcome</div>
3   </template>
4
5   <script>
6   export default {
7     name: 'HomeComponent',
8     data: () => ({
9
10    })
11
12  }
13  </script>
```

Nội dung do người
dùng tùy chỉnh

Tạo trang hiển thị danh sách

1. Tạo thủ tục

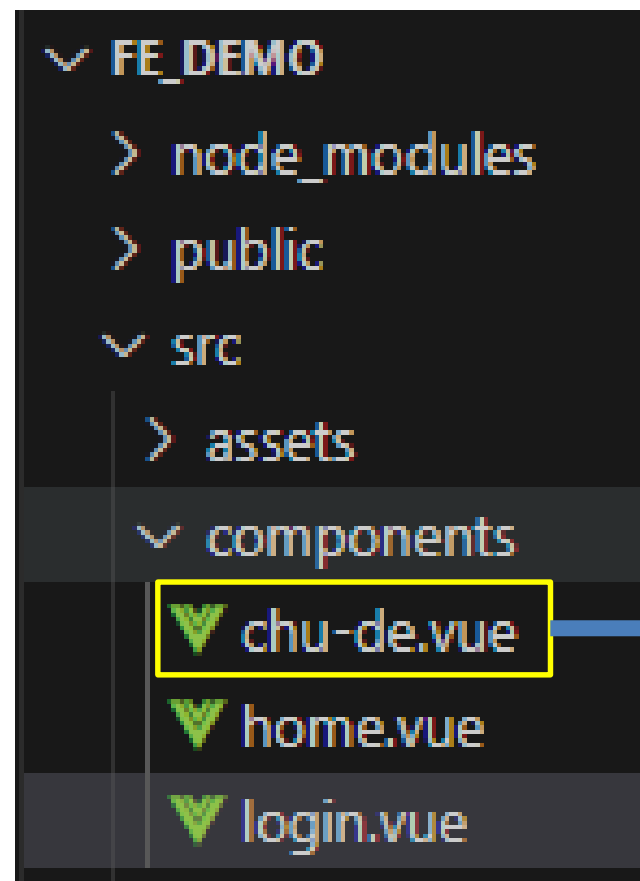


```
- create proc [dbo].[spDanhMuc_ChudeSelectALL]  
as  
    select * from tbl_Chude
```

2. Tạo trang login



Tạo file `chu-de.vue` trong thư mục components của dự án



```
<template>
  <div class="text-center my-3">
    <div class="mb-2"><b>DANH SÁCH CHỦ ĐỀ</b></div>
    <v-btn elevation="0" color="success">
      <v-icon class="mr-2">mdi-plus-circle</v-icon>
      Thêm mới
    </v-btn>
  </div>
  <v-table style="width: 400px" class="mx-auto">
    <thead>
      <tr>
        <th>Mã CD </th>
        <th class="text-left">Tên chủ đề </th>
        <th>Sửa </th>
        <th>Xoá </th>
      </tr>
    </thead>
    <tbody>
      <tr v-for="item in dataChuDe" :key="item.maCD">
        <td>{{ item.MaCD }}</td>
        <td>{{ item.TênChuDe }}</td>
        <td><v-icon>mdi-pencil</v-icon></td>
        <td><v-icon>mdi-delete</v-icon></td>
      </tr>
    </tbody>
  </v-table>
</template>

<script>
import axios from '../utils/axios'
export default {
  name: 'ChuDeComponent',
  data: () => ({
    dataChuDe: []
  }),
  mounted() {
    this.getChuDe()
  },
  methods: {
    getChuDe() {
      axios.get('/chude', {})
        .then((response) => { this.dataChuDe = [...response.data.data] })
        .catch((error) => {
          console.log(error);
        });
    }
  }
}
</script>
```

Giao diện trang chủ-de

Vòng lặp để hiển thị tất cả các dòng trong bảng chủ đề

Dữ liệu từ api trả về

Gọi api lấy danh sách chủ đề, nếu thành công thì dung biến **dataChuDe** để chứa danh sách mà api trả về

Câu hỏi thảo luận ???