

Vue 後台 backoffice 專案開發01

林小強

outline

- 下載現有admin範本網站
- 建立ecpro01專案
- 後台目錄說明
- 使用elment-ui前端框架
- 修改登入/view/login/index. vue
- 調整login登入驗證方式與註解
- 調整store/modules/user. js

使用現有範本網站

- <https://github.com/PanJiaChen/vue-admin-template>

sts > test001.txt

```
1 //下載使用現有範本網站
2 git clone https://github.com/PanJiaChen/vue-admin-template
3
4 //更改目錄名稱
5 rename vue-admin-template backoffice01
6
7 //進入backoffice01目錄
8 cd backoffice01
9
10 //安裝相依模組
11 npm install
12
13 //測試啟動
14 npm run dev
```

測試執行

Login Form

admin

.....眼睛

Login

username: admin password: any

後台目錄說明1

2. 目錄說明：

build

.....----index.js webpack配置文件，很少會修改此文件

mock

.....----mock數據的文件夾，模擬的數據使用mock.js來產生的

node_modules

.....----專案項目相依的各類模組

public

.....----icon圖示，靜態頁面，public文件夾裡面，經常放置一些靜態資源，webpack封裝時，不會編譯這個文件夾，原封不動的打包到dist文件夾裡面。

src

.....----程式碼來源區

.....----api文件夾：設計涉及到遠端請求相關，

.....----assets文件夾：裡面放置一些靜態資源(一般共享)，放在assets文件裡面靜態資源，在webpack打包的時候，會進行編譯。

.....----components文件夾：一般放置非路由組件獲取全局組件

.....----icons：這個文件夾的裡面放置一些svg向量圖

.....----layout文件夾：裡面放置一些版面配置的元件如siderbar與navbar等

後台目錄說明2

src

-----程式碼來源區

-----api文件夾：設計涉及到遠端請求相關，

-----assets文件夾：裡面放置一些靜態資源(一般共享)，放在assets文件裡面靜態資源，在webpack打包的時候，會進行編譯。

-----components文件夾：一般放置非路由組件獲取全局組件

-----icons：這個文件夾的裡面放置一些svg向量圖

-----layout文件夾：裡面放置一些版面配置的元件如siderbar與navbar等

-----store文件夾：vuex的程式區

-----style文件夾：與樣式相關，存放網站的樣式表

-----utils文件夾：專案所自訂開發的工具程式區

-----views文件夾：裡面放置的是路由組件

tests

-----測試目錄區

app.vue：啟動root根vue組件

main.js：網站進入的js檔

permission.js與權限相關

setting系統配置配

后台範本使用element-ui前端框架



搜索文档

指南

组件

主题

资源

2.15.8

中文

国际化

自定义主题

内置过渡动画

组件

Basic

Layout 布局

Container 布局容器

Color 色彩

Typography 字体

Border 边框

Icon 图标

Button 按钮

Link 文字链接

Form

Radio 单选框

Button 按钮

常用的操作按钮。

基础用法

基础的按钮用法。

The screenshot shows a grid of 20 button components arranged in four rows. The first row contains six buttons labeled: '默认按钮' (Default), '主要按钮' (Primary), '成功按钮' (Success), '信息按钮' (Info), '警告按钮' (Warning), and '危险按钮' (Danger). The second row contains six buttons labeled: '朴素按钮' (Plain), '主要按钮' (Primary), '成功按钮' (Success), '信息按钮' (Info), '警告按钮' (Warning), and '危险按钮' (Danger). The third row contains five buttons labeled: '圆角按钮' (Rounded), '主要按钮' (Primary), '成功按钮' (Success), '信息按钮' (Info), and '危险按钮' (Danger). The fourth row contains five icons: a magnifying glass, a blue circular arrow, a green checkmark, a grey envelope, and an orange star.

显示代码

在线运行

平台設定setting.js

The screenshot shows a code editor interface with a sidebar on the left and a main editor area on the right. The sidebar lists various files and folders under '已開啟的編輯器' and 'BACKOFFICE01'. In the main editor, the file 'settings.js' is open. Three specific lines of code are highlighted with red boxes:

```
1 module.exports = {  
2   ...  
3   title: '後台管理系統',  
4   ...  
5   /**  
6    * @type {boolean} true | false  
7    * @description Whether fix the header  
8    * navbar是否設定為固定  
9    */  
10  fixedHeader: true,  
11  ...  
12  /**  
13   * @type {boolean} true | false  
14   * @description Whether show the logo in sidebar  
15   * sidebar是否顯示Logo列  
16   */  
17  sidebarLogo: true  
18}  
19
```

The highlighted lines define configuration options: 'title' (set to '後台管理系統'), 'fixedHeader' (set to true), and 'sidebarLogo' (set to true).

平台設定setting.js1

移至(G) 執行(R) 終端機(T) 說明(H) settings.js - backoffice03 - Visual Studio Code

src > settings.js > <unknown> > ecPlatForm

```
3   title: '後台管理系統',
4
5 > /**
6   fixedHeader: true,
7
8 > /**
9   sidebarLogo: true,
10 > // sidebarLogo: false,
11 >
12 > /**
13   ecPlatForm: {
14     // db_Url: 'http://front.edu/lesson-14/No_attache/backoffice03/public/getdata/',
15     db_Url: '',
16     // ec_Url: 'http://front.edu/lesson-14/No_attache/',
17     ec_Url: '',
18     images: 'images',
19     productImages: 'product_img',
20   },
21
22
23
24
25
26
27 }
28
29
```

測試執行

Vue Admin Template

Dashboard / Example / Table

≡

Dashboard Example Table Tree Form Nested External Link

ID	Title	Author	Pageviews	Status	Display_time
0	Esupzwzvr nsahoes kizw dwy yxjv dfyuf kflfbkdee zrjcevgd xwg ynn pthru ckgpz nuttfqnaix tpowx yhhncjew.	name	3371	draft	⌚1988-12-09 03:11:28
1	Qldl ogoeg ofcxriw uiv lqmopbnwyg iyihblfxos fkzadjv kaene ffuhrag blas ioblku hrpuox kdprwintq p mkboj kfgrxpal fewllrcba vpxvzbu olzhuxnb qhvmi wqyzetua.	name	3418	draft	⌚1993-09-15 20:13:19
2	Sfxod ghtgxs ium virlcnlwv roip tqducnyd uuujqcni rytiogggd ylseudxjgi fficw nqtc dp ktjyhtml qzy ssh qzmmggecv grhuuioeb mmpeiqfw.	name	3872	deleted	⌚1975-11-28 15:53:03
3	Gtodmcoo ssjxbdq poekl degpxvnae fhwtctif ehzhcfgky iwrupydl medsutswr zmufhn elew miurt rotglo xkh ibph.	name	4965	published	⌚2010-05-04 23:04:24
4	Yfgpwli qzrmo oswbdi qqzefhz nfbhx yefvglufy owrpgm eaut jwrfcw fwxg.	name	4713	published	⌚1999-07-17 11:48:15
5	Yonuwx edot ydnizk hnpvbx pucdkqhu klwbncaqd yqmgilo stvbgxrho jpnli qerxb mquuxo rhmtuh t bhszq etqxtwjjt tjb.	name	2342	draft	⌚2007-05-10 21:52:50

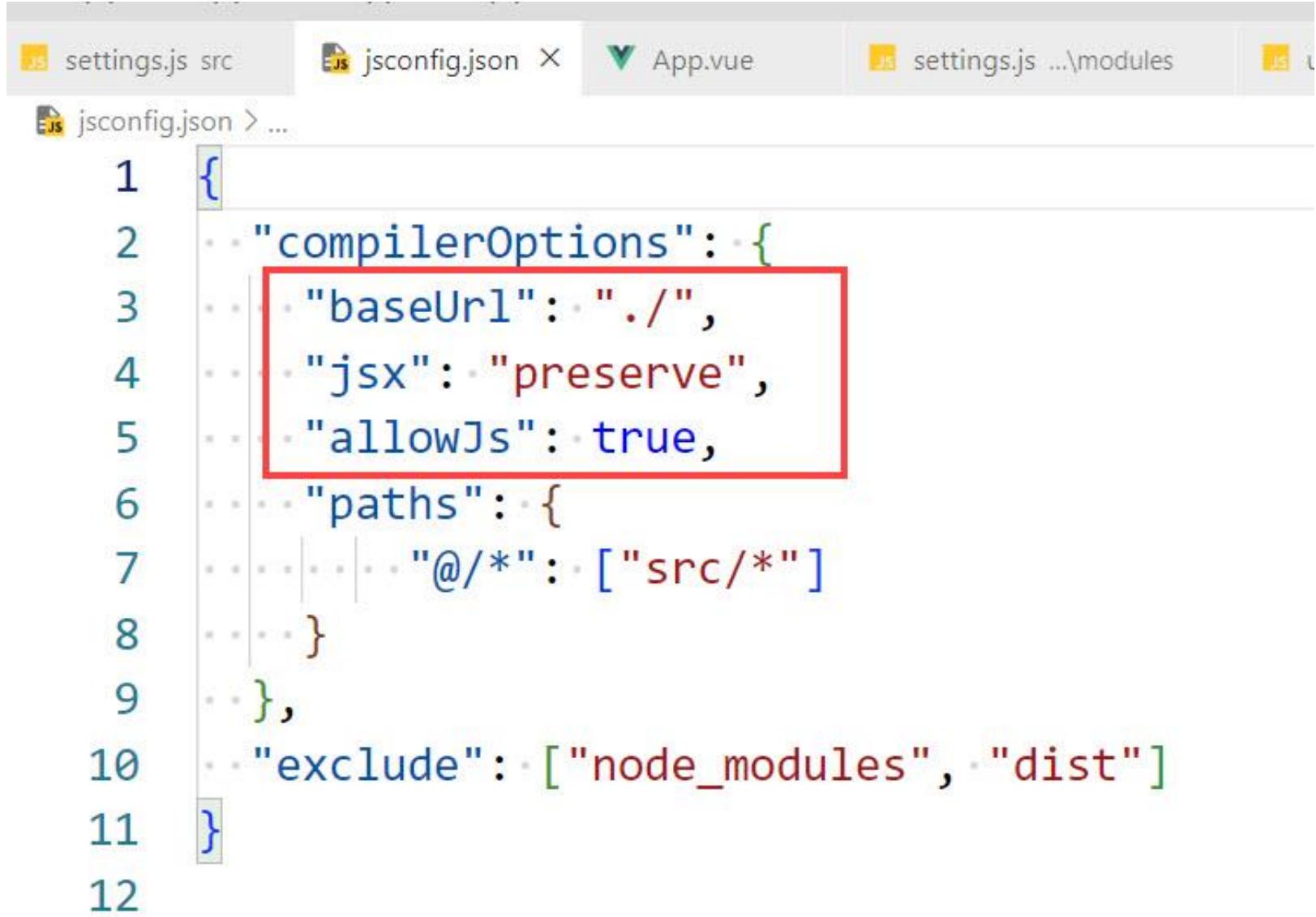
開啟view/login/index.vue

The screenshot shows a code editor interface with the following details:

- File Structure:** The left sidebar shows a project structure:
 - views
 - BACKOFFICE01
 - icons
 - layout
 - router
 - store
 - styles
 - utils
 - views
 - form
 - login
 - index.vue
 - nested
 - table
 - tree
 - 404.vue
 - App.vue
 - main.js
 - permission.js
 - settings.js
 - tests
 - .editorconfig
 - .env.development
 - .env.production
 - .env.staging
 - .eslintignore
 - .eslintrc.js
 - .gitignore
- Current File:** The main editor area shows the content of `index.vue`. The file is a Vue template with the following structure:

```
<template>
  <div class="login-container">
    <el-form ref="loginForm" :model="loginForm" :rules="loginRules" class="login-form"
      auto-complete="on" label-position="left">
      <div class="title-container">
        <h3 class="title">Login Form</h3>
      </div>
      <el-form-item prop="username">
        <span class="svg-container">
          <svg-icon icon-class="user" />
        </span>
        <el-input
          ref="username"
          v-model="loginForm.username"
          placeholder="Username">
      </el-form-item>
    </el-form>
  </div>
```
- Editor Tools:** At the bottom, there are tabs for 問題 (1), 輸出, 偵錯主控台, and 終端機. A tooltip message is displayed: "TypeScript intellisense is disabled on template. To enable, configure `"javascriptConfigFile"` in the `"compilerOptions"` property of tsconfig or jsconfig. To disable this prompt instead, config...".

修正 JSX 問題



The screenshot shows a code editor interface with several tabs at the top: "settings.js src", "jsconfig.json", "App.vue", "settings.js ...\\modules", and another "jsconfig.json" tab. The "jsconfig.json" tab is currently active, displaying the following JSON configuration:

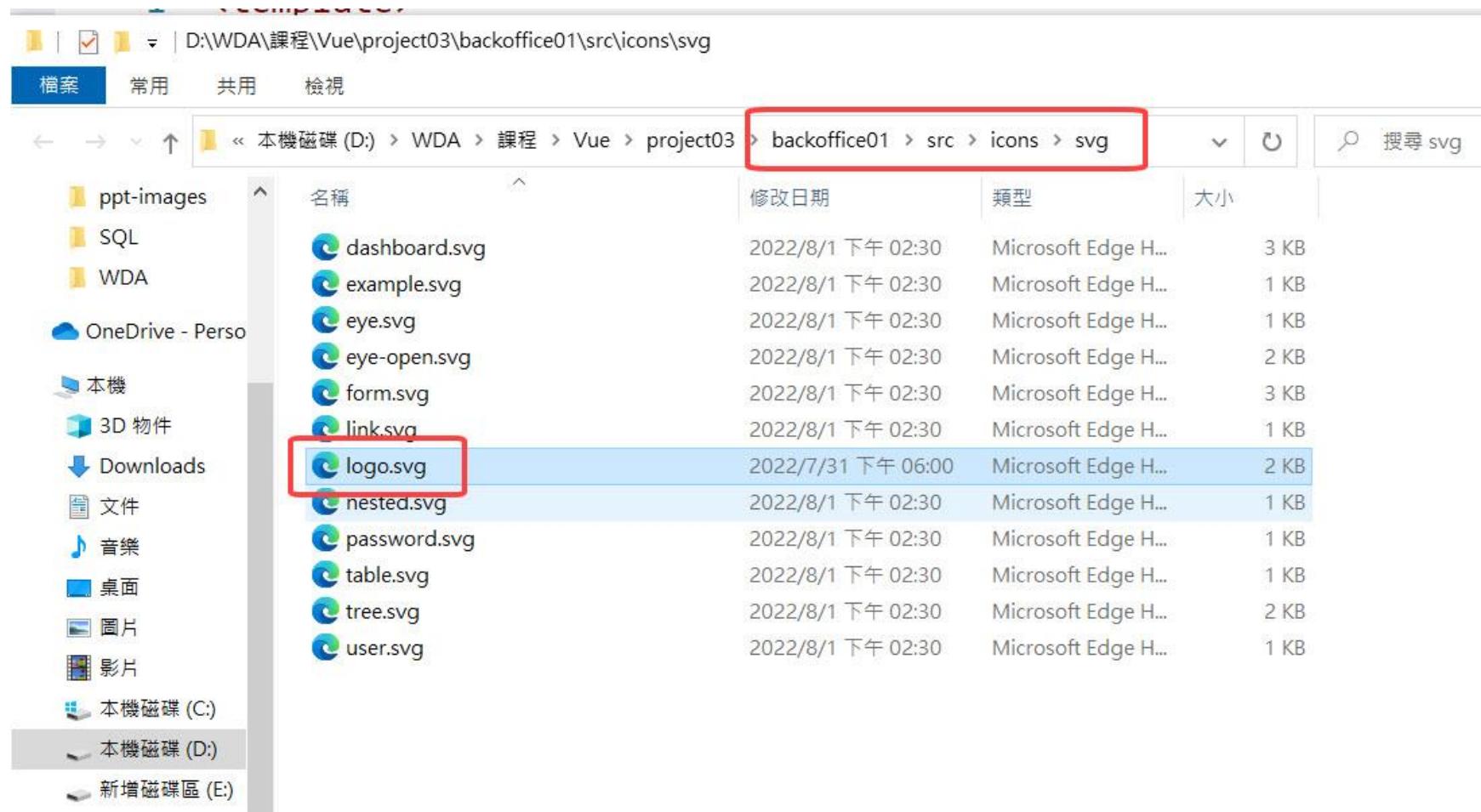
```
1  {
2    "compilerOptions": {
3      "baseUrl": "./",
4      "jsx": "preserve",
5      "allowJs": true,
6      "paths": {
7        "@/*": ["src/*"]
8      }
9    },
10   "exclude": ["node_modules", "dist"]
11 }
```

A red rectangular box highlights the following section of the code:

```
3      "jsx": "preserve",
4      "allowJs": true,
```

This highlights the configuration for JSX and JavaScript interop.

Src/icons/svg放入logo.svg圖



Src/Views/login/index.vue 建立 logo.svg 程式

```
<div class="login-container">
  <el-form ref="loginForm" :model="loginForm" :rules="loginRules" class="login-form">
    <div style="text-align: center;">
      <svg-icon icon-class="logo" class="logo"></svg-icon>
    </div>
    <div class="title-container">
      <h3 class="title">管理者登入</h3>
    </div>
    <el-form-item prop="username">
      <span class="svg-container">
        <svg-icon icon-class="user" />
      </span>
      <el-input v-model="loginForm.username" placeholder="用户名" type="text" />
    </el-form-item>
    <el-form-item prop="password">
      <span class="svg-container">
        <svg-icon icon-class="lock" />
      </span>
      <el-input v-model="loginForm.password" placeholder="密码" type="password" />
    </el-form-item>
    <div class="button-group">
      <el-button type="primary" @click="handleLogin">登 录 </el-button>
      <el-button @click="handleReset">重置 </el-button>
    </div>
  </el-form>
</div>
```

Src/Views/login/index.vue 建立 logo.svg 程式

```
...<div style="text-align: center;">
...  <svg-icon icon-class="logo" class="logo" />
...</div>
```

```
...<div class="title-container">
...  <h3 class="title">管理者登入</h3>
...</div>
```

變更登入樣式

```
1 reference
184 .login-container {
185   min-height: 100%;
186   width: 100%;
187   background: linear-gradient(□ □#fff, ■ ■#2263bf);
188   overflow: hidden;
189
1 reference
190   .login-form {
191     position: relative;
192     width: 420px;
193     max-width: 100%;
194     padding: 35px 0 0 35px;
195     margin: 0 auto;
196     overflow: hidden;
197     background-color: $bg;
198     margin-top: 2%;
199 }
```

建立logo類別樣式

```
219 ... }
```

1 reference

```
220 .logo {  
221   width: 80px;  
222   height: 80px;  
223   margin-bottom: 15px;  
224 }
```

測試執行



測試修改程式，檢查帳號名稱是否正確，與密碼長度

```
name: 'Login',
data() {
  // 此為element-ui自訂功能，為檢查帳號名稱是否正確，與密碼長度是否大於六位以上
  const validateUsername = (rule, value, callback) => {
    if (!validUsername(value)) {
      callback(new Error('請輸入正確的帳號！！'))
    } else {
      callback()
    }
  }
  const validatePassword = (rule, value, callback) => {
    if (value.length < 6) {
      callback(new Error('密碼長度不可小於6個字元以下。'))
    } else {
      callback()
    }
  }
  return {
    loginForm: {

```

測試執行



調整登入驗證方式與註解

```
  },
  loginRules: {
    // 檢查帳號是否存在
    // username: [{ required: true, trigger: 'blur', validator: validateUsername }],
    username: [{ required: true, trigger: 'blur', }],
    // 檢查密碼是否大於6位元
    password: [{ required: true, trigger: 'blur', validator: validatePassword }]
  },
  loading: false,
```

註解登入程式

```
},
handleLogin() {
  // 此為帳號與密碼，進行伺服器發出需求，是否符合規則
  this.$refs.loginForm.validate(valid => {
    //如果符合登入驗證規則
    if (valid) {
      this.loading = true; //開啟按鈕的loading的效果
      //產生一個vuex的action在user/login進行帳號與密碼檢查
      this.$store.dispatch('user/login', this.loginForm)
        .then(() => {
          //帳號密碼檢查通過，登入成功，進行路由的跳轉
          this.$router.push({ path: this.redirect || '/' })
          this.loading = false //結束按鈕的loading的效果
        }).catch(() => {
          this.loading = false
        })
    } else {
      console.log('error submit!!!')
      return false
    }
  })
}
```

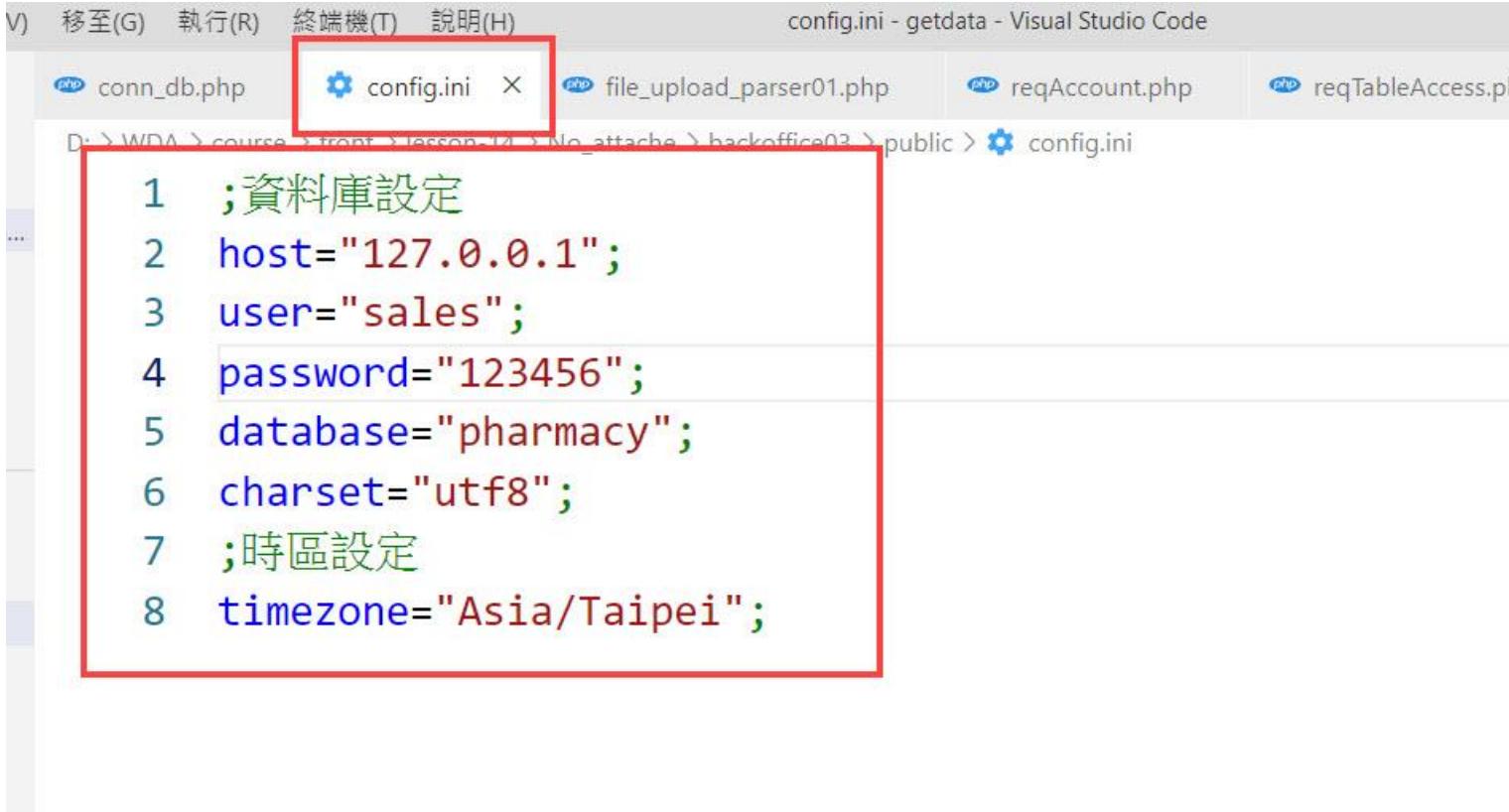
先取消util/request.js的X-Token

```
14 // request interceptor
15 // 伺服器請求攔截器，在發出伺服器前，預先執行程式
16 service.interceptors.request.use(
17   config => {
18     // do something before request is sent
19     if (store.getters.token) {
20       // let each request carry token
21       // ['X-Token'] is a custom headers key
22       // please modify it according to the actual situation
23       config.headers['X-Token'] = getToken() // 取消在header加入token
24     }
25     return config
26   },
27   error => {
28     // do something with request error
29     console.log(error) // for debug
30     return Promise.reject(error)
31   }
)
```

修改request伺服器回應攔截器

```
// response interceptor
// 伺服器回應攔截器，在收到伺服器時，先執行的程式
service.interceptors.response.use(
  ...
  ...
  response => {
    const res = response.data
    // if the custom code is not 20000, it is judged as an error.
    //伺服器回應失敗處理訊息
    if (res.code != 20000 && res.code != 200) {
      Message({
        message: res.message || 'Error',
        type: 'error',
        duration: 5 * 1000
      })
      // 50008: Illegal token; 50012: Other clients logged in; 50014: Token expired;
      if (res.code === 50008 || res.code === 50012 || res.code === 50014) {
        // to re-login
        MessageBox.confirm('You have been logged out, you can cancel to stay on this')
      }
    }
  }
)
```

建立public/config.ini



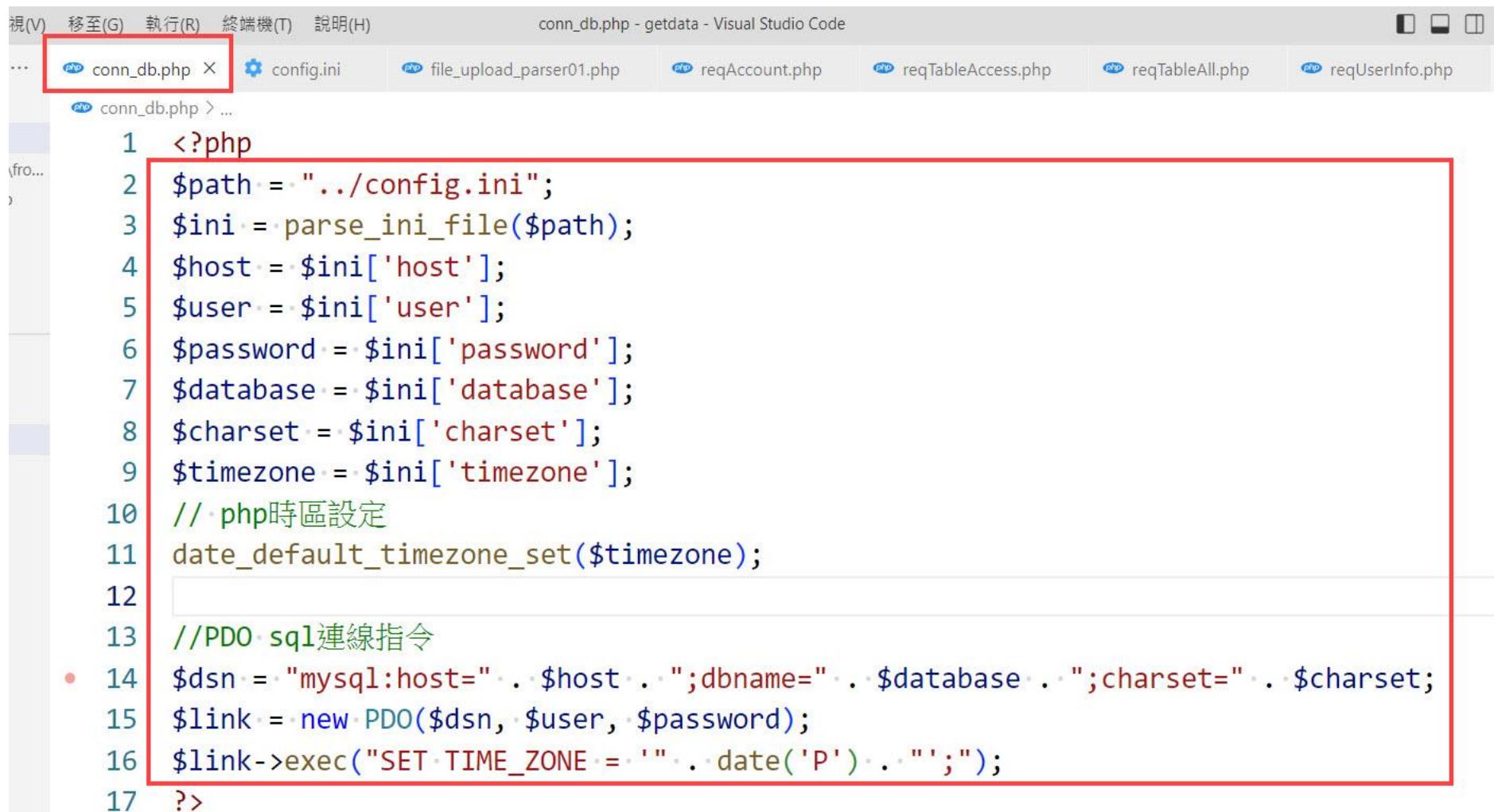
config.ini - getdata - Visual Studio Code

D:\WDA\course>front>lesson_12>No_attach>backoffice03>public>config.ini

```
1 ;資料庫設定
2 host="127.0.0.1";
3 user="sales";
4 password="123456";
5 database="pharmacy";
6 charset="utf8";
7 ;時區設定
8 timezone="Asia/Taipei";
```

建立/getdata/conn_db.php環境

- 建立資料庫連線



```
conn_db.php - getdata - Visual Studio Code
conn_db.php X config.ini file_upload_parser01.php reqAccount.php reqTableAccess.php reqTableAll.php reqUserInfo.php
conn_db.php > ...

1 <?php
2 $path = "../config.ini";
3 $ini = parse_ini_file($path);
4 $host = $ini['host'];
5 $user = $ini['user'];
6 $password = $ini['password'];
7 $database = $ini['database'];
8 $charset = $ini['charset'];
9 $timezone = $ini['timezone'];
10 // php時區設定
11 date_default_timezone_set($timezone);
12
13 //PDO sql連線指令
14 $dsn = "mysql:host=" . $host . ";dbname=" . $database . ";charset=" . $charset;
15 $link = new PDO($dsn, $user, $password);
16 $link->exec("SET TIME_ZONE = '" . date('P') . "'");
```

後端管理者資料庫

	adid 管理員 ID	adlogin 管理員 登入名	adname 管理員 姓名	active 是否活 動	adlevel 管理員 級別	groupid 管理員 群組	avatar 頭像	adpasswd 管理員密 碼	ademail 管理員email	allowip 允許存取IP	lastlogin 管理員最後登 入時間	sip 管理員 IP	sbrowser 管理員 瀏覽器	splatform 管理員最 後登入 OS	create_date 建立日期	update_date 修改日期	re 備
<input type="checkbox"/>	1	phil	林小強1	1	99	1	pic02.svg	e10adc3949ba59abbe56e057f20f883e	phil@gmail.com		2022-07-31 03:56:49	127.0.0.1	Chrome	Windows 10	2018-11-29 16:10:21	2022-07-31 09:56:49	N
<input type="checkbox"/>	2	test01 01	測試帳號 01	0	2	2	pic01.gif	e10adc3949ba59abbe56e057f20f883e	phil02@gmail.com	192.168.50.1,192.168.60.1,192.168.70.2, 192.168.8...	2019-01-11 00:00:00	NULL		2019-01-11 10:58:01	2022-08-05 11:44:57	N	
<input type="checkbox"/>	3	a1a1 a1a1	測試帳號 a1a1	1	1	1	pic01.gif	123456	a1a1@gmail.com	192.168.50.1,192.168.60.1, 192.168.70.1	2019-03-11 08:46:03	127.0.0.1	Chrome	Windows 7	2019-01-11 11:24:23	2022-08-05 11:45:24	N
<input type="checkbox"/>	4	a1a2 a1a2	測試帳號 a1a2	1	1	1	pic02.svg	e10adc3949ba59abbe56e057f20f883e	a1a1@gmail.com	192.168.50.1	2019-03-11 08:46:03	NULL		2019-01-11 11:35:14	2022-08-04 14:29:26	N	

↑ 全選 已選擇項目：

全部顯示 資料列數： 25 選擇資料列： 搜尋此資料表 依主鍵排序：

建立reqAccount.php帳號確認1

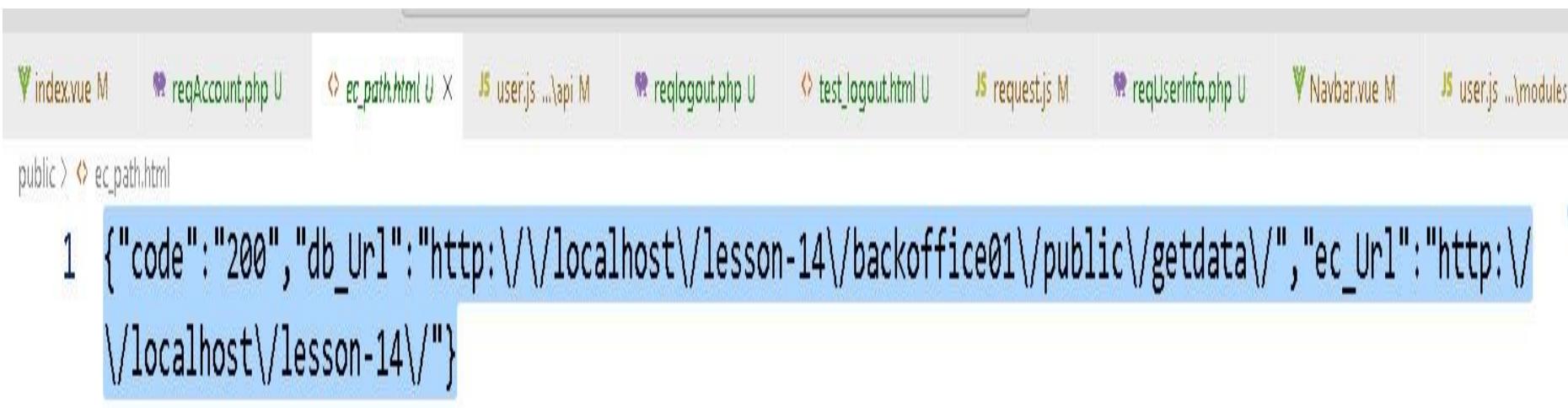
```
1 <?php
2 header('Access-Control-Allow-Origin:*');
3 header('Content-Type: application/json;charset=utf-8'); //return json string
4
5 require_once('conn_db.php'); //取得資料連線設定
6 if (isset($_POST['username']) && isset($_POST['password'])) {
7     $username = $_POST['username'];
8     $password = $_POST['password'];
9     //進行帳號密碼查詢
10    $SQLstr = sprintf("SELECT ad_user.*,ad_group.gpname,ad_group.gpename, ad_group.active as gActive FROM ad_user,ad_group
11 WHERE ad_user.groupid=ad_group.groupid AND adlogin='%s' AND adpasswd='%s' ", $username, $password);
12    $result = $link->query($SQLstr);
13    if ($result && $result->rowCount() != 0) {
14        $data = $result->fetch();
15        if ($data['active'] == 1) {
16            (!isset($_SESSION)) ? session_start() : '';
17            $_SESSION['token'] = session_id(); //使用session編碼為token值
18            $_SESSION['adid'] = $data['adid'];
19            $_SESSION['adlogin'] = $data['adlogin'];
20            $_SESSION['adname'] = $data['adname'];
21            $_SESSION['ademail'] = $data['ademail'];
22            $_SESSION['groupid'] = $data['groupid'];
23            $_SESSION['gpname'] = $data['gpname'];
```

建立reqAccount.php帳號確認2

```
21     $_SESSION['groupid'] = $data['groupid'];
22     $_SESSION['gpname'] = $data['gpname'];
23     $_SESSION['gename'] = $data['gename'];
24     $_SESSION['avatar'] = $data['avatar'];
25     $_SESSION['lastlogin'] = $data['lastlogin'];
26     $_SESSION['allowip'] = $data['allowip'];
27     $_SESSION['sip'] = $data['sip'];

28
29     $retcode = array("code" => "20000", "data" => array("token" => $_SESSION['token'], ), "message" => "success",
30     "status" => "OK");
31 } else {
32     $retcode = array("code" => "50002", "data" => "", "message" => "帳號或是群組被停用，請連絡管理人員！！", "status"
33     => "ERROR");
34 }
35 } elseif ($result->rowCount() == 0) {
36     $retcode = array("code" => "50006", "data" => "", "message" => "帳號或密碼錯誤，請重新輸入！！", "status" => "ERROR");
37 } else {
38     $retcode = array("code" => "40002", "data" => "", "message" => "抱歉！資料無法取得後台資料庫，請連絡管理人員。",
39     "status" => "ERROR");
40 }
41 echo json_encode($retcode, JSON_UNESCAPED_UNICODE);
42
43 return;
```

建立Public/ec_path.html 電商平台 路徑設定



The screenshot shows a code editor with multiple tabs at the top, including index.vue, reqAccount.php, ec_path.html (which is the active tab), user.js...api, reqLogout.php, test_logout.html, request.js, reqUserInfo.php, Navbar.vue, and user.js...modules. Below the tabs, the file content is displayed:

```
public > ec_path.html
1 {"code": "200", "db_Url": "http://localhost\lesson-14\backoffice01\public\getdata\", "ec_Url": "http://localhost\lesson-14\"}
```

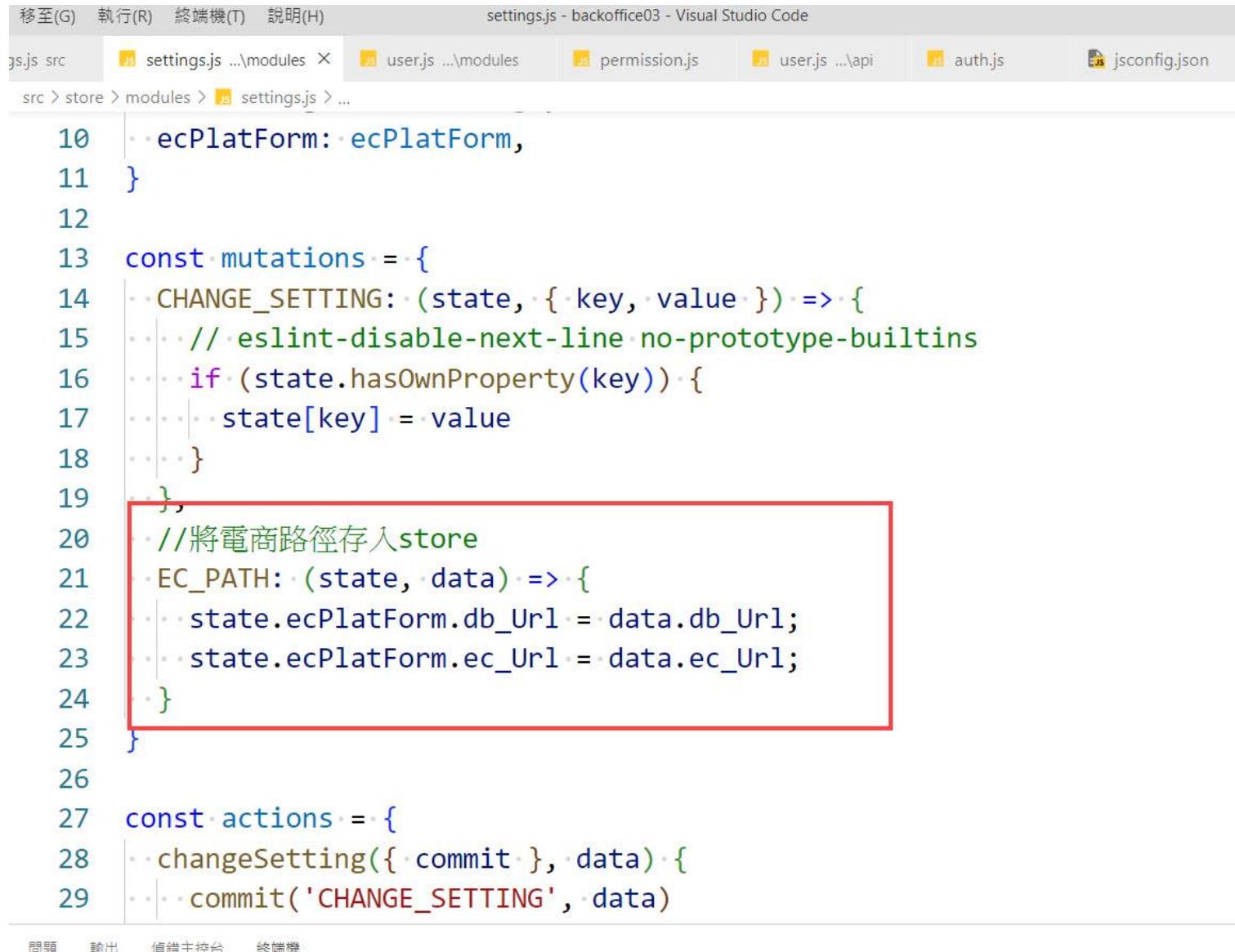
修改store/modules/settings.js

```
h.html U JS user.js ...\\api M reqlogout.php U test_logout.html U JS request.js M reqUserInfo.php U Navbar.vue M JS user.js ...\\modules M  
: > store > modules > JS settings.js  
1 import request from '@/utils/request';
2 import {getEcUrl, setEcUrl} from '@/utils/auth'
3 import defaultSettings from '@/settings'
4
5 const { showSettings, fixedHeader, sidebarLogo, ecPlatform } = defaultSettings
6 const state = {
7   showSettings,
8   fixedHeader,
9   sidebarLogo,
10  ecPlatform: ecPlatform,
11 }
12 const mutations = {
13   CHANGE_SETTING: (state, { key, value }) => {
14     // eslint-disable-next-line no-prototype-builtins
15     if (state.hasOwnProperty(key)) {
16       state[key] = value
17     }
18   },
19   // 將電商路徑存入store
20   EC_PATH: (state, data) => {
21     state.ecPlatform.db_Url = data.db_Url;
22     state.ecPlatform.ec_Url = data.ec_Url;
23   }
24 }
```

新增setting.js action模組程式

```
27 const actions = {
28   changeSetting({ commit }, data) {
29     commit('CHANGE_SETTING', data)
30   },
31   //在./public目錄取得ec_path.html路徑設定檔
32   async ec_Path({ commit }) {
33     let result = await request({ url: `./ec_path.html`, method: 'GET' });
34     if (result.code === 200) {
35       commit('EC_PATH', result); //將路徑存入store
36       setEcUrl(result); //將路徑存入cookies
37       return true;
38     }
39   },
40   //在路徑有存入cookies，直接取出，存入store
41   getEc_Cookies({ commit }) {
42     let result = { db_Url: getEcUrl("db_Url"), ec_Url: getEcUrl("ec_Url") };
43     commit('EC_PATH', result);
44     return true;
45   }
46 }
```

新增setting Mutations模組程式



settings.js - backoffice03 - Visual Studio Code

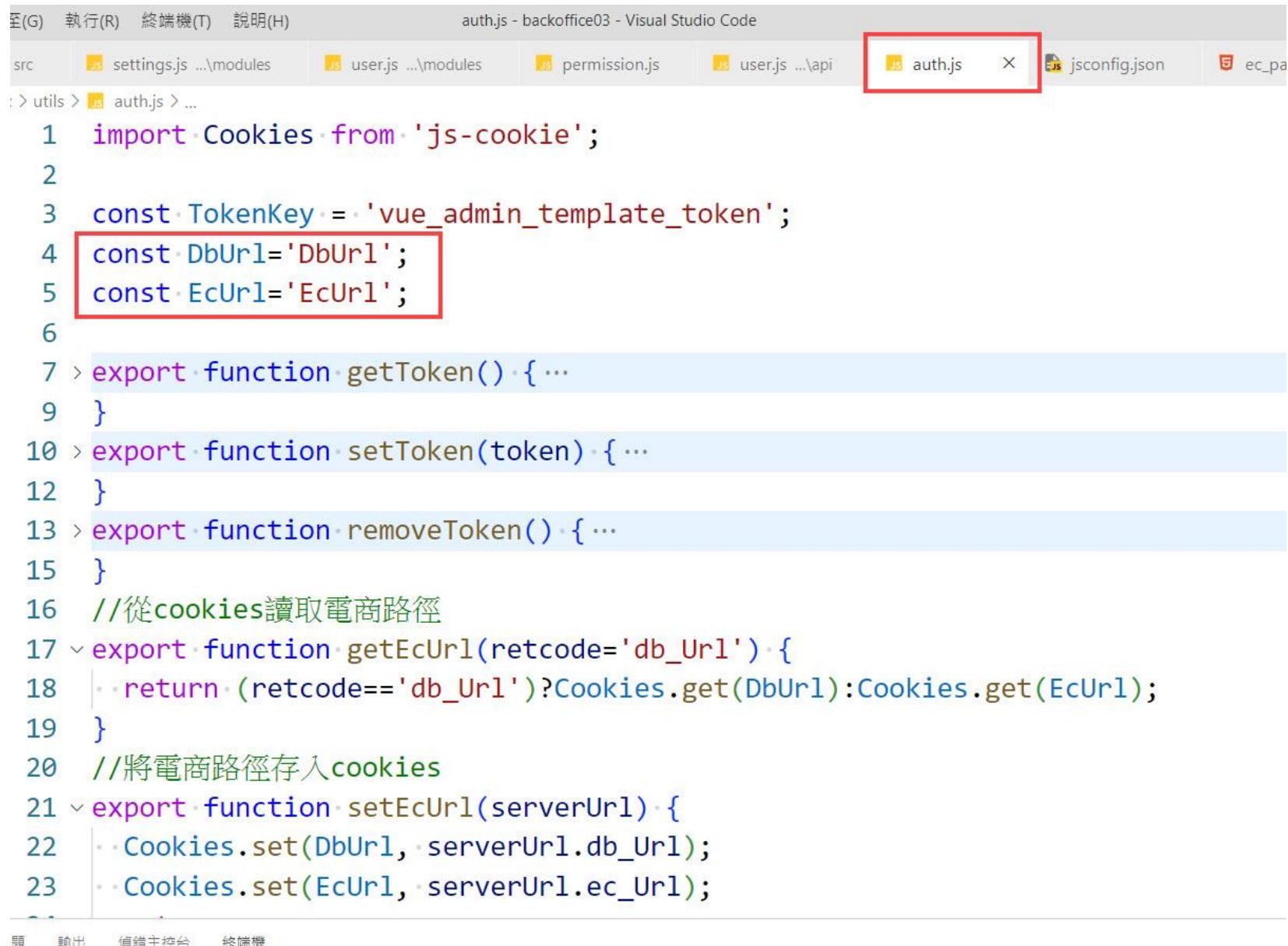
```
js.js src settings.js ...\\modules user.js ...\\modules permission.js user.js ...\\api auth.js jsconfig.json
```

src > store > modules > settings.js > ...

```
10  .. ecPlatForm: ecPlatForm,
11  }
12
13 const mutations = {
14  .. CHANGE_SETTING: (state, { key, value }) => {
15    // eslint-disable-next-line no-prototype-builtins
16    if (state.hasOwnProperty(key)) {
17      state[key] = value
18    }
19  },
20  // 將電商路徑存入store
21  EC_PATH: (state, data) => {
22    state.ecPlatForm.db_Url = data.db_Url;
23    state.ecPlatForm.ec_Url = data.ec_Url;
24  }
25}
26
27 const actions = {
28  changeSetting({ commit }, data) {
29    commit('CHANGE_SETTING', data)
```

問題 輸出 偵錯主控台 終端機

新增util/auth.js cookies程式



```
至(G) 執行(R) 終端機(T) 說明(H) auth.js - backoffice03 - Visual Studio Code
src settings.js ...\\modules user.js ...\\modules permission.js user.js ...\\api auth.js X jsconfig.json ec_pa
: > utils > auth.js > ...
1 import Cookies from 'js-cookie';
2
3 const TokenKey = 'vue_admin_template_token';
4 const DbUrl='DbUrl';
5 const EcUrl='EcUrl';
6
7 > export function getToken() { ...
9 }
10 > export function setToken(token) { ...
12 }
13 > export function removeToken() { ...
15 }
16 //從cookies讀取電商路徑
17 < export function getEcUrl(retcode='db_Url') {
18   return (retcode=='db_Url')?Cookies.get(DbUrl):Cookies.get(EcUrl);
19 }
20 //將電商路徑存入cookies
21 < export function setEcUrl(serverUrl) {
22   Cookies.set(DbUrl, serverUrl.db_Url);
23   Cookies.set(EcUrl, serverUrl.ec_Url);
.
```

新增util/auth.js cookies程式1

```
14
15 > export function removeToken() { ...
16 }
17
18 //從cookies讀取電商路徑
19 export function getEcUrl(retcode='db_Url') {
20   let cookData;
21   switch (retcode){
22     case 'db_Url':
23       cookData=Cookies.get(DbUrl);
24       break;
25     case 'ec_Url':
26       cookData=Cookies.get(EcUrl);
27       break;
28   };
29   return (cookData);
30 }
31 //將電商路徑存入cookies
32 export function setEcUrl(serverUrl) {
33   debugger;
34   Cookies.set(DbUrl, serverUrl.db_Url);
35   Cookies.set(EcUrl, serverUrl.ec_Url);
36   return ;
37 }
38 //從cookies移除電商路徑
```

新增util/auth.js cookies程式1

```
28 | ... };
29 | ...
30 | }
31 | //將電商路徑存入cookies
32 | export function setEcUrl(serverUrl) {
33 |   debugger;
34 |   Cookies.set(DbUrl, serverUrl.db_Url);
35 |   Cookies.set(EcUrl, serverUrl.ec_Url);
36 |   return ;
37 | }
38 | //從cookies移除電商路徑
39 | export function removeEcUrl() {
40 |   Cookies.remove(DbUrl);
41 |   Cookies.remove(EcUrl);
42 |   return ;
43 | }
44 | 
```

Store/modules/user.js新增變數1

```
1 import { loginApi, logoutApi, getInfoApi } from '@/api/user'
2 import { getToken, setToken, removeToken, removeEcUrl } from '@/utils/auth'
3 import {MD5} from '@/utils/commlib'
4 import { resetRouter } from '@/router'
5
6 const getDefaultState = () => {
7   return {
8     token: getToken(), // 登入後從伺服器取得token
9     name: '', // 管理者名稱
10    avatar: '', // 頭像圖檔
11    person: [], // 管理者登入後個人資料存放區
12  }
13}
14
15 const state = getDefaultState()
16
17 const mutations = {
18   RESET_STATE: (state) => {
19     Object.assign(state, getDefaultState())
20   },
21 }
```

Store/modules/user.js新增變數2

```
17 const mutations = {
18   RESET_STATE: (state) => {
19     Object.assign(state, getDefaultstate())
20   },
21   SET_TOKEN: (state, token) => {
22     state.token = token
23   },
24   SET_NAME: (state, name) => {
25     state.name = name
26   },
27   SET_AVATAR: (state, avatar) => {
28     state.avatar = avatar
29   },
30   // 加入管理員資訊寫入state
31   SET_PERSON: (state, person) => {
32     state.person = person;
33   },
34 }
35
```

Store/modules/user.js修改程式

```
--  
36 const actions = {  
37     // login()為處理管理者帳密到伺服器認證請求  
38     async login({ commit }, userInfo) {  
39         const { username, password } = userInfo; // 取得登入頁的管理者帳密  
40         //呼叫src/api/user.js->loginApi()驗證帳號與密碼  
41         let result = await loginApi({ username: username.trim(), password: MD5(password) });  
42         if (result.code == 20000) { // 當取得返回code==20000時為帳密驗證成功  
43             //取回帳密確認成功，將TOKEN資料取回填入store欄位  
44             commit('SET_TOKEN', result.data.token);  
45             setToken(result.data.token);  
46             return 'success';  
47         } else {  
48             console.log(result);  
49             return Promise.reject(new Error('faile'));  
50         }  
51     },  
52     // get user info  
53     async getInfo({ commit, state }) {  
54 }
```

Src/api/user.js修改程式

The screenshot shows a code editor with multiple tabs at the top: 'igs.js src', 'user.js ...\\api X', 'settings.js ...\\modules', 'user.js ...\\modules', 'permission.js', 'auth.js', and 'jsconfig.json'. Below the tabs, the file structure is shown as 'src > api > user.js > getInfoApi'. The main code area contains the following JavaScript code:

```
1 import request from '@/utils/request'
2
3 //對後端資料庫發出帳密驗證函數
4 export function loginApi(data) {
5   return request({
6     url: 'reqAccount.php',
7     method: 'POST',
8     data: "username="+data.username+"&password="+data.password,
9   });
10  // return request({
11  //   url: '/vue-admin-template/user/login',
12  //   method: 'post',
13  //   data
14  // })
15}
16
```

A red rectangular box highlights the entire block of code starting from line 3, which contains the logic for logging in to a backend database using 'reqAccount.php'. This highlights the specific part of the code being modified.

修改util/request.js要求攔截器模組程式

```
5  
6 // create an axios instance  
7 const service = axios.create({  
8   // baseURL: process.env.VUE_APP_BASE_API, // url = base.url + request.url  
9   baseURL: '', .....//建立預設後端路徑  
10  timeout: 25000 // request timeout  
11 })  
12  
13 // request interceptor  
14 // 伺服器請求攔截器，攜帶的token字段  
15 service.interceptors.request.use(  
16   config => {  
17     //判斷store.getters.db_Url是否有存電商EC_dbUrl路徑，有則採用，無則空  
18     if (store.getters.db_Url) {  
19       config.baseURL = store.getters.db_Url;  
20     }  
21     //如果store已經有寫入token表示  
22     if (store.getters.token) {  
23       // debugger;  
24       // let each request carry token  
25       // 'Token' is a custom header key
```

修改store/getter.js程式

The screenshot shows the Visual Studio Code interface with the following details:

- File menu: 修复(G), 打开(O), 终端(T), 帮助(H)
- Tab bar: settings.js src, request.js, getters.js (active), user.js ...\\api, settings.js ...\\modules, user.js ...\\modules
- Breadcrumb: src > store > getters.js > getters > db_Url
- Code content (getters.js):

```
1 const getters = {
2   sidebar: state => state.app.sidebar,
3   device: state => state.app.device,
4   token: state => state.user.token,
5   avatar: state => state.user.avatar,
6   name: state => state.user.name,
7   // 增加管理者個人資訊到getters
8   person: state => state.user.person,
9   db_Url: state => state.settings.ecPlatform.db_Url,
10  ec_Url: state => state.settings.ecPlatform.ec_Url,
11}
12 export default getters
13
```

The code block from line 7 to line 10 is highlighted with a red border.

新增permission.js 程式

The screenshot shows a code editor with the file `permission.js` open. The code implements a router guard using `router.beforeEach` to handle progress bars, set page titles, and manage database URLs.

```
4 import NProgress from 'nprogress' // progress bar
5 import 'nprogress/nprogress.css' // progress bar style
6 import { getToken, getEcUrl } from '@/utils/auth' // get token from cookie
7 import getPageTitle from '@/utils/get-page-title'
8
9 NProgress.configure({ showSpinner: false }) // NProgress Configuration
10 const whiteList = ['/login'] // no redirect whitelist 預設執行頁面/login
11
12 // 使用路由守衛router.beforeEach在每次路由進行跳轉時，即會產生檢查程式執行
13 router.beforeEach(async (to, from, next) => {
14   // start progress bar
15   NProgress.start()
16
17   // set page title
18   document.title = getPageTitle(to.meta.title)
19   // debugger;
20   // 建立電商平台的資料庫路徑，並判斷是否有存入Cookies，無則在public讀入ec_path.html，有則直接從cookies將路徑放入store
21   const dbName = (store.getters.dbName === undefined) ? "" : store.getters.dbName;
22   if (dbName === "") {
```

The code is annotated with several red boxes:

- A red box surrounds the import statements for `NProgress`, `getToken`, and `getEcUrl`.
- A red box surrounds the entire `router.beforeEach` block.
- A red box surrounds the explanatory comment above the `router.beforeEach` block.
- A red box surrounds the `if (dbName === "") {` condition at the bottom.

新增permission.js程式1

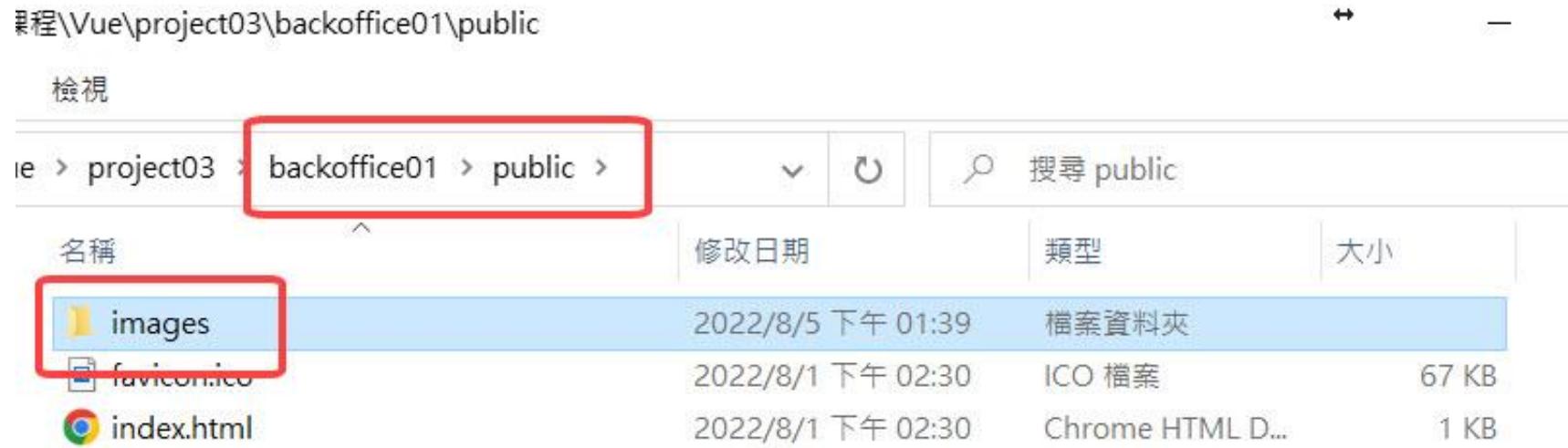
```
16  
17 // set page title  
18 document.title = getPageTitle(to.meta.title)  
19 // debugger;  
20 //建立電商平台的資料庫路徑，並判斷是否有存入Cookies，無則在public讀入ec_path.html，有則直接從cookies將路徑放入store  
21 const getDb_Url = (store.getters.db_Url==undefined)?"":store.getters.db_Url;  
22 if (getDb_Url=="") {  
23   if (getEcUrl("db_Url") == undefined) {  
24     await store.dispatch("settings/ec_Path");  
25   } else {  
26     await store.dispatch("settings/getEc_Cookies");  
27   }  
28 }  
29  
30
```

測試執行登入

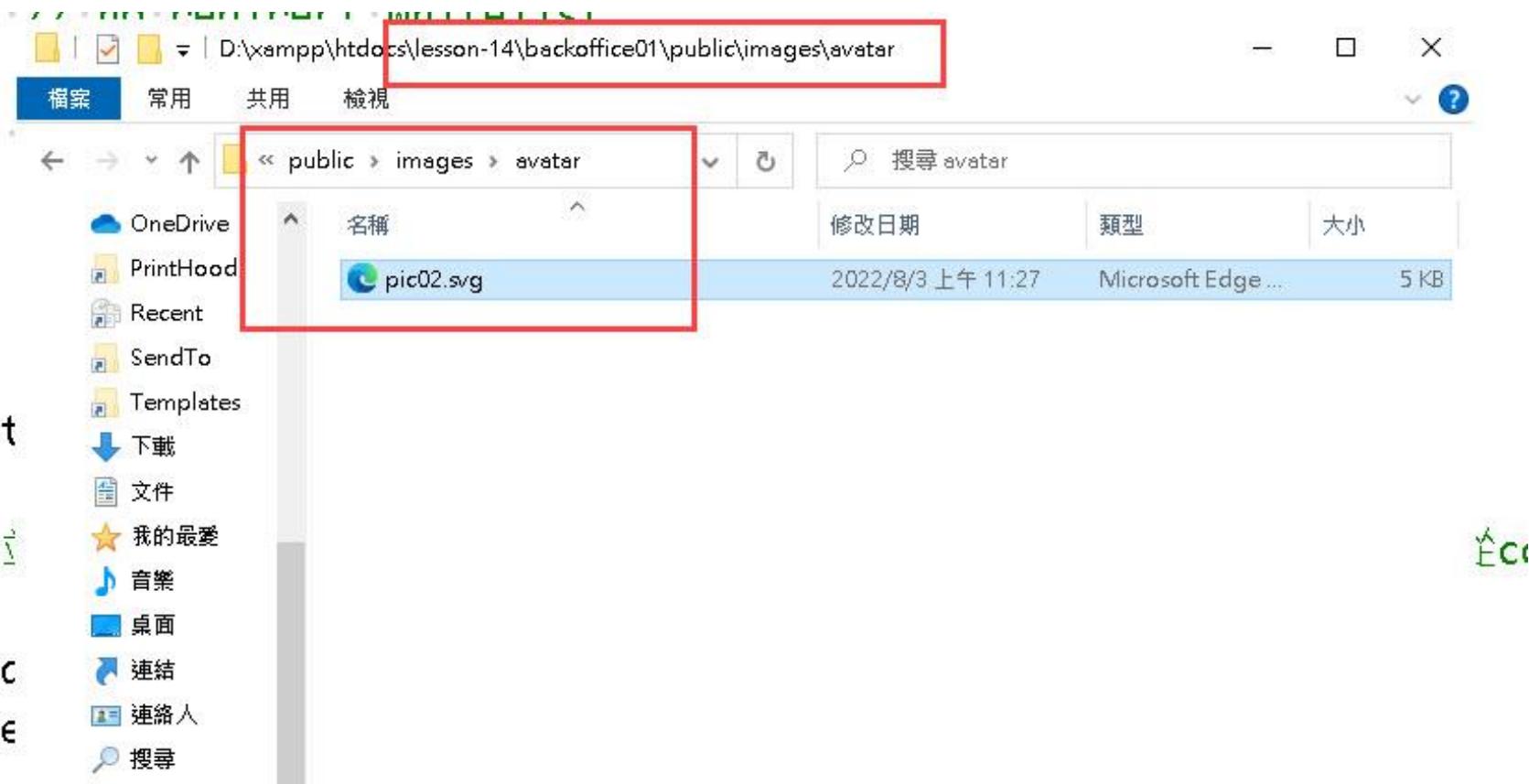


A screenshot of a Vue Admin Template dashboard. The left sidebar contains navigation items: Dashboard, Example (with a dropdown arrow), Form, Nested (with a dropdown arrow), and External Link. The main content area shows a user profile with the name "phil". In the top right corner, there is a user icon (a person icon inside a square) with a dropdown arrow. A red box highlights this user icon.

複製images圖示到public



複製images圖示到public



修改 /layout/components/Navbar.vue

```
<div class="right-menu">
  <el-dropdown class="avatar-container" trigger="click">
    <div class="avatar-wrapper">
      <!-- 
      
      <i class="el-icon-caret-bottom" />
    </div>
    <el-dropdown-menu slot="dropdown" class="user-dropdown">
      <router-link to="/">
        <el-dropdown-item>回首頁
        </el-dropdown-item>
      </router-link>
      <el-dropdown-item divided @click.name="handleNameClick">
        <span style="display:block;">系統名稱</span>
      </el-dropdown-item>
    </el-dropdown-menu>
  </el-dropdown>
</div>
```

```
39  export default {
40    components: {
41      Breadcrumb,
42      Hamburger
43    },
44    computed: {
45      ...mapGetters([
46        'sidebar',
47        'avatar',
48        'name', 加入
49      ])
50    },
51  },
```

測試執行



系統重整時發生錯誤

The screenshot shows the Vue DevTools interface. The top navigation bar includes tabs for Elements, Vue (which is selected), Console, and Sources. Below the tabs is a toolbar with icons for back, forward, search, and refresh. The main area displays the application structure under 'App 1' version '2.6.10'. The 'Root' component has a child 'app' component with a 'namespaced' configuration. Under 'state', there is a 'user' object and a 'person' array. The 'Console' tab in the bottom panel shows several warning messages from the Vue runtime, indicating that properties like 'dangerouslyUseHTMLString', 'message', and 'showClose' must be accessed via '\$data' due to proxying restrictions.

```
vue.runtime.esm.js?2b0e:619
must be accessed with "$data.dangerouslyUseHTMLString" because properties
starting with "$" or "_" are not proxied in the Vue instance to prevent
conflicts with Vue internalsSee: https://vuejs.org/v2/api/#data

(found in <Root>)
vue.runtime.esm.js?2b0e:619
with "$data.message" because properties starting with "$" or "_" are not
proxied in the Vue instance to prevent conflicts with Vue internalsSee: http
s://vuejs.org/v2/api/#data

(found in <Root>)
vue.runtime.esm.js?2b0e:619
accessed with "$data.showClose" because properties starting with "$" or "_"
are not proxied in the Vue instance to prevent conflicts with Vue
internalsSee: https://vuejs.org/v2/api/#data

(found in <Root>)
```

處理資料重整permission.js

```
JS main.js      ▼ Navbar.vue    JS permission.js X  JS auth.js      JS request.js    ▼ index.vue    JS user.js ...\\api   JS user.js ...\\modu
src > JS permission.js > ⚡ router.beforeEach() callback
  5 import 'nprogress/nprogress.css' // progress bar style
  6 import { getToken } from '@/utils/auth' // get token from cookie
  7 import getPageTitle from '@/utils/get-page-title'
  8
  9 NProgress.configure({ showSpinner: false }) // NProgress Configuration
 10
 11 const whiteList = ['/login'] // no redirect whitelist 預設執行頁面/login
 12 // 使用路由守衛router.beforeEach在每次路由進行跳轉時，即會產生檢查程式執行
 13 router.beforeEach(async(to, from, next) => {
 14   ...
 15 })
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
```

處理資料重整permission.js

```
const hasToken = getToken() //試著取得token
if (hasToken) {
  if (to.path === '/login') {
    // if is logged in, redirect to the home page
    next({ path: '/' })
    NProgress.done()
  } else {
    const hasGetUserInfo = store.getters.name //從store取得管理者資料
    if (hasGetUserInfo) {
      next()
    } else {
      try {
        // get user info getInfo，若管理者資料遺失，則重新呼叫/user/getInfo讀取
        await store.dispatch('user/getInfo')
        next()
      } catch (error) { ... }
    }
  }
}
```

修改/store/modules/user.js

```
// get user info
//改寫為async/await的方式，進行伺服器的請求使用者資料
async getInfo({ commit, state }) {
  let result = await getInfoApi(state.token); // 當返回code==20000時為token驗證成功
  if (result.code == 20000) {
    //token確認成功，將資料取回填入store欄位
    commit('SET_PERSON', result.data);
    commit('SET_NAME', result.data.adlogin);
    commit('SET_AVATAR', result.data.avatar);
    return 'success';
  } else {
    return Promise.reject(new Error(result.message));
  }
  // return new Promise((resolve, reject) => {
  // })
},

```

對映的 import 修改

```
c > store > modules > JS user.js > actions > getInfo > result
1 import { loginApi, logout, getInfoApi } from '@/api/user'
2 import { getToken, setToken, removeToken } from '@/utils/auth'
3 import { MD5 } from '@/utils/commlib'; // 使用自訂函數MD5
4 import { resetRouter } from '@/router'
5
6 const getDefaultState = () => {
7   return {
8     token: getToken(), // 登入後從伺服器取得token
9     name: '' // 管理者名稱

```

修改/src/api/user.js

建立reqUserInfo.php管理資訊1

```
heckAccount.php    ↗ reqlogout.php    ↗ reqUserInfo.php X    ↗ reqAccount.php    ↗ test001.html  
reqUserInfo.php > ...  
1 <?php  
2 header('Access-Control-Allow-Origin:*');  
3 header('Content-Type: application/json; charset=utf-8'); //return json string  
4  
5 if(isset($_GET['token']) && $_GET['token']!=''){  
6     $token = $_GET['token'];  
7     session_id($token);  
8     session_start();  
9     if($_SESSION['token']==$token){  
10        $retcode = array("code"=>"20000", "data"=>array(  
11            "token"=>$_SESSION['token'],  
12            "adid"=>$_SESSION['adid'],  
13            "adlogin"=>$_SESSION['adlogin'],  
14            "adname"=>$_SESSION['adname'],  
15            "ademail"=>$_SESSION['ademail'],  
16            "groupid"=>$_SESSION['groupid'],  
17            "gpname"=>$_SESSION['gpname'],  
18            "gename"=>$_SESSION['gename'],  
19            "avatar"=>$_SESSION['avatar'],  
20        ), "message"=>"success", "status"=>"OK");  
21    }  
22}
```

建立reqUserInfo.php管理資訊2

```
..... "gpname"=>$_SESSION['gpname'],
..... "gename"=>$_SESSION['gename'],
..... "avatar"=>$_SESSION['avatar'],
..... ),"message"=>"success","status"=>"OK");
}else{
    $retcode =array("code"=>"50008","data"=>"","message"=>"token無效，請重新登入！！",
    "status"=>"Fail");
}
echo json_encode($retcode, JSON_UNESCAPED_UNICODE);
}
return;
?>
```

測試執行

The screenshot shows the Vue DevTools Inspector interface. At the top, there's a navigation bar with tabs for Elements, Vue, Console, Sources, and more. Below the navigation bar, there's a search bar labeled "Find apps..." and a dropdown menu showing "App 1" and its version "2.6.10".

The main area displays a hierarchical state tree. The root node is "Root", which has three children: "app", "settings", and "user". The "user" node is expanded, showing its properties: "token", "name", "avatar", and "person". The "person" object is further expanded, showing properties like "ademail", "adid", "adlogin", "adname", "avatar", "gename", "gpname", "groupid", and "token". A red box highlights this expanded "person" object.

At the bottom of the screen, there are tabs for "Console", "What's New", and "Issues". The "Console" tab is selected. It contains a message from WDS (Webpack Dev Server) indicating it's waiting for an update signal. There are also some warning messages about source maps and file not found.

連接前台路徑

Vue Admin Template

Dashboard

Example

首頁輪播管理

Table

Tree

Form

Nested

Menu1

Menu1-1

Menu1-2

Menu1-2-1

Menu1-2-2

Dashboard

管理者登入資訊

登入帳號: test

管理者email: test@gmail.com

管理者ID: 10

管理者姓名: 林小強2

管理者頭像: pic02.svg 

管理者群組: 系統管理人員

最後登入時間:

回首頁

前台首頁

系統登出

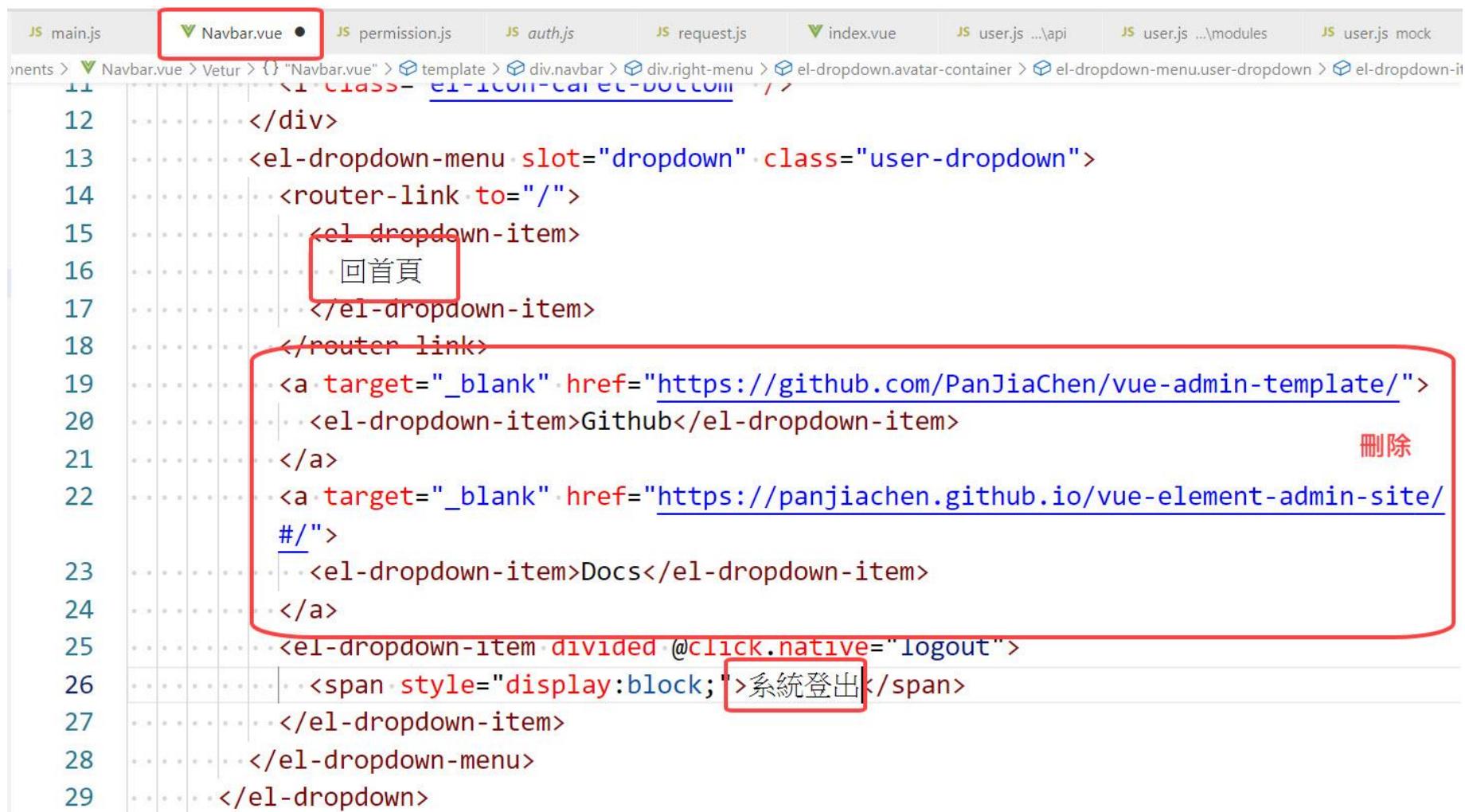
在這裡輸入文字來搜尋

Windows Start button

任务栏图标: File Explorer, Edge, Chrome, Chat, Mail, Calendar, Photos

系统托盘: Volume, Language, Screen, Date/Time (上午 10:49, 2025/3/5)

修改 /src/layout/components/navbar.vue



```
JS main.js JS Navbar.vue • JS permission.js JS auth.js JS request.js JS index.vue JS user.js ...api JS user.js ...modules JS user.js mock

components > Navbar.vue > Vetur > {} "Navbar.vue" > template > div.navbar > div.right-menu > el-dropdown.avatar-container > el-dropdown-menu.user-dropdown > el-dropdown-item

11           class="el-icon-caret-bottom" />
12       </div>
13       <el-dropdown-menu slot="dropdown" class="user-dropdown">
14         <router-link to="/">
15           <el-dropdown-item>
16             回首頁
17           </el-dropdown-item>
18         </router-link>
19         <a target="_blank" href="https://github.com/PanJiaChen/vue-admin-template/">
20           <el-dropdown-item>Github</el-dropdown-item>
21         </a>
22         <a target="_blank" href="https://panjiachen.github.io/vue-element-admin-site/#/">
23           <el-dropdown-item>Docs</el-dropdown-item>
24         </a>
25         <el-dropdown-item divided @click.native="logout">
26           <span style="display:block;">系統登出</span>
27         </el-dropdown-item>
28       </el-dropdown-menu>
29     </el-dropdown>
```

建立前台標籤程式

```
<el-dropdown>
  <el-dropdown-menu slot="dropdown" class="user-dropdown">
    <router-link to="/">
      <el-dropdown-item>
        回首頁
      </el-dropdown-item>
    </router-link>
    <el-dropdown-item divided>
      <a style="display:block;" v-bind:href="frontUrl" target="_blank">前台首頁
      </a>
    </el-dropdown-item>
    <el-dropdown-item divided @click.native="logout">
      <span style="display:block;">系統登出</span>
    </el-dropdown-item>
  </el-dropdown-menu>
</el-dropdown>
```

取得前台路徑資料

```
38 export default {
39   data(){
40     return {
41       //前台URL路徑
42       frontUrl: '',
43     }
44   },
45   components: { ... },
46 },
47 computed: { ... },
48 },
49 mounted() {
50   //電商前台專URL路徑
51   this.frontUrl = this.$store.state.settings.ecPlatform.ec_Url;
52 },
53 methods: {
54   toggleSideBar() {
55 }
```

處理系統登出作業

The screenshot shows a Vue Admin Template dashboard. On the left is a sidebar with a dark blue background and white text. It includes a logo, the title "Vue Admin Template", and several menu items: "Dashboard", "Example", "Form", "Nested", and "External Link". The main content area has a light gray background and displays the text "name: phil". In the top right corner, there is a user profile icon with a green and blue gradient. To its right is a vertical ellipsis menu. A red rectangular box highlights a dropdown menu on the far right, which contains four items: "Home", "Github", "Docs", and "Log Out".

- Home
- Github
- Docs
- Log Out

修改Navbar.vue的JS程式

```
30 import Breadcrumb from '@/components/Breadcrumb'  
31 import Hamburger from '@/components/Hamburger'  
32 import { Message } from 'element-ui';  
33 export default {  
34   components: { ...  
35     },  
36   computed: { ...  
37     },  
38   methods: {  
39     toggleSideBar() {  
40       this.$store.dispatch('app/toggleSideBar')  
41     },  
42     async logout() {  
43       let result = await this.$store.dispatch('user/logout');  
44       if(result.code == 20000){ ... //若code=20000則成功登出  
45         Message({type: 'success', message: result.message});  
46       }  
47       this.$router.push(` /login?redirect=${this.$route.fullPath}`);  
48     }  
49   }  
50 }
```

修改/store/modules/user.js

```
94 // user logout
95 //使用者登出頁面，預計清除session資料, token資料
96 async logout({ commit, state }) {
97   let result=await logoutApi(state.token);
98   if(result.code=="20000"){
99     removeToken(); ..... // must remove token first
100    resetRouter(); ..... // 重設系統路由器
101    commit('RESET_STATE'); .. //重設store的所有變數
102    return(result);
103  } else {
104    return Promise.reject(new Error(result.message));
105  }
106 // return new Promise((resolve, reject) => { ...
107 // })
116 },
117
```

修改 import

> store > modules > **JS** user.js > ...

```
1 import { loginApi, logoutApi, getInfoApi } from '@/api/user'  
2 import { getToken, setToken, abc logoutApi  
3 import { MD5 } from '@/utils/commlib'; // 使用自訂函數MD5  
4 import { resetRouter } from '@/router'  
5  
6 const getDefaultState = () => {  
7   return {  
8     token: getToken(), // 登入後從伺服器取得token  
9     name: '', // 管理者名稱  
10    avatar: '', // 頭像圖檔
```

修改/src/api/user.js

```
18 |  
19 | //對伺服器發出登出的功能函數  
20 | export function logoutApi(token) {  
21 |   debugger;  
22 |   return request({  
23 |     url: 'reqlogout.php',  
24 |     method: 'post',  
25 |     data: "token="+token,  
26 |   })  
27 | }  
28 |
```

建立reqlogout.php登出程式1

checkAccount.php

reqlogout.php X

reqUserInfo.php

reqAccount.php

test001.html

reqlogout.php > ...

```
1 <?php
2 header('Access-Control-Allow-Origin:*');
3 header('Content-Type: application/json;charset=utf-8'); //return json string
4
5 if(isset($_POST['token']) && $_POST['token']!=''){
6     $token = $_POST['token'];
7     session_id($token);
8     session_start();
9     if($_SESSION['token']==$token){
10         unset($_SESSION['token']);
11         unset($_SESSION['adid']);
12         unset($_SESSION['adlogin']);
13         unset($_SESSION['adname']);
14         unset($_SESSION['ademail']);
15         unset($_SESSION['groupid']);
16         unset($_SESSION['gpname']);
17         unset($_SESSION['gpname']);
18         unset($_SESSION['avatar']);
```

建立reqlogout.php登出程式1

```
..... unset($_SESSION['gename']);  
..... unset($_SESSION['avatar']);  
..... unset($_SESSION['lastlogin']);  
..... unset($_SESSION['sip']);  
..... $retcode =array("code"=>"20000","data"=>"", "message"=>"系統成功登出！！", "status"=>"OK");  
..... setcookie(session_name(), '', time() - 42000,);  
..... session_destroy();  
} else{  
..... $retcode =array("code"=>"50008", "data"=>"", "message"=>"token無效，請重新登入！！",  
..... "status"=>"Fail");  
}  
..... echo json_encode($retcode, JSON_UNESCAPED_UNICODE);  
}  
return;  
?>
```

執行測試



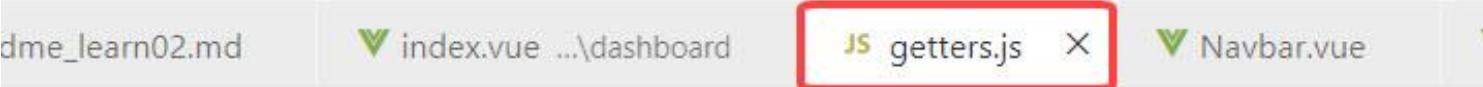
修改/views/dashboard/index.vue

The screenshot shows a code editor interface with the following details:

- File Structure:** The left sidebar shows a tree view of files and folders. Under "src", "views", and "dashboard", there are "index.vue" files. Other files like "Navbar.vue", "permission.js", and "request.js" are also listed.
- Current File:** The main editor area is displaying the content of "index.vue".
- Code Content:**

```
22 export default {
23   name: "Dashboard",
24   computed: {
25     ...mapGetters(["name", "person"]),
26   },
27 };
28 </script>
29
30 <style lang="scss" scoped>
31   .dashboard {
32     &-container {
33     }
34     &-text {
35       font-size: 30px;
36       line-height: 46px;
37     }
38     p{
39       font-size: 14px;
40       line-height: 1.5em;
41     }
42   
```
- Annotations:** Two specific sections of the code are highlighted with red boxes:
 - A red box surrounds the line `...mapGetters(["name", "person"])`.
 - A red box surrounds the entire `p{` block from line 38 to line 41.
- Editor Header:** The top bar shows tabs for "readme_learn02.md", "index.vue", "Navbar.vue", "index.vue", and "JS". It also displays the current file path: "src > views > dashboard > index.vue" and the line number "22".

修改/store/getters.js



The screenshot shows a code editor interface with several tabs at the top: 'dme_learn02.md', 'index.vue ...\\dashboard', 'JS getters.js' (which is highlighted with a red box), and 'Navbar.vue'. Below the tabs, the file structure is shown as ': > store > JS getters.js > [e] getters'. The main content area contains the following code:

```
1 const getters = {
2   sidebar: state => state.app.sidebar,
3   device: state => state.app.device,
4   token: state => state.user.token,
5   avatar: state => state.user.avatar,
6   name: state => state.user.name,
7   // 增加管理者個人資訊到getters
8   person: state => state.user.person,
9 }
10 export default getters
11
```

A red box highlights the line '// 增加管理者個人資訊到getters' and the entire block from line 7 to line 8.

修改template

```
1 <template>
2   <div class="dashboard-container">
3     <div class="dashboard-text">
4       管理者登入資訊
5       <hr>
6       <p>登入帳號 : {{ name }}</p>
7       <p>管理者email : {{ person.ademail }}</p>
8       <p>管理者ID : {{ person.adid }}</p>
9       <p>管理者姓名 : {{ person.adname }}</p>
10      <p>管理者頭像 : {{ person.avatar }}</p>
12      <p>管理者群組 : {{ person.gpname }}</p>
13      <p>最後登入時間 : {{ person.lastlogin }}</p>
14      <p>最後登入IP : {{ person.sip }}</p>
15      <p>取得伺服器token : {{ person.token }}</p>
16    </div>
17  </div>
18 </template>
```

執行測試

Vue Admin Template

Dashboard

Example

Form

Nested

External Link

Dashboard

管理者登入資訊

登入帳號: phil

管理者email: phil@gmail.com

管理者ID: 1

管理者姓名: 林小強1

管理者頭像: pic02.svg 

管理者群組: 系統管理人員

最後登入時間:

最後登入IP:

取得伺服器token: 29bc3bga37vbiggt0c6pmp7q9

/views/login/index.vue

```
43     .....<span class="show-pwd" @click="showPwd">
44     .....<svg-icon :icon-class="passwordType === 'password' ? 'eye' : 'eye-open'" />
45     .....
```

執行測試



第一階段完成

- 完成範本網頁的建置
- 完成backoffice管理者登入對接資料庫
- 完成backoffice管理者資料更新
- 完成backoffice管理者登出
- 更新navbar管理者avatar圖像
- 完成backoffice管理者在dashboard個人資料呈現。
- 最後完成login畫面功能的修訂