

## Postman Create User And Update to MongoDB

The image displays a development environment with three main components:

- VS Code Editor:** Shows the `authController_06.js` file with the following code:

```
1 import User_06 from "../models/User_06.js";
2
3 const register_06 = async (req, res) => {
4   console.log('body', req.body);
5   const user = await User_06.create(req.body);
6   res.status(201).json({ user });
7   // res.send('register user -- Tony Zhan , 2074105');
8 }
9
10 const login_06 = async (req, res) => {
11   res.send('Login user -- Tony Zhan , 2074105');
12 }
13 const updateUser_06 = async (req, res) => {
14   res.send('update user -- Tony Zhan , 2074105');
15 }
16
```

  
The terminal output shows:

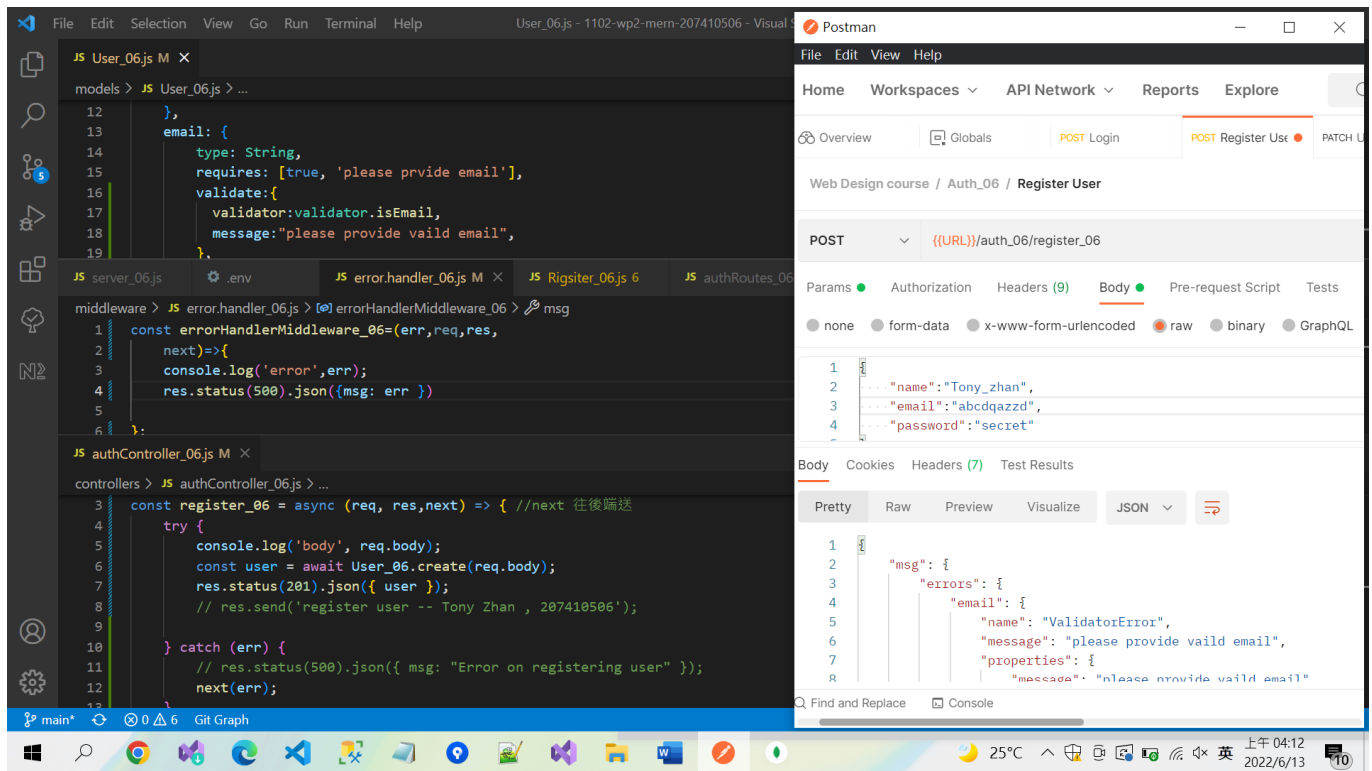
```
keyValue: { email: 'abcdqazzaq@gmail.com' },
[Symbol(errorLabels)]: Set(0) {}
[nodemon] app crashed - waiting for file changes before
[nodemon] restarting due to changes...
[nodemon] starting 'node server_06.js'
Connectin to MongoDB
Server is running on port 5000
body {
  name: 'Tony_zhan',
  email: 'abcdqazzaq@gmail.com',
  password: 'secret'
}
```
- Postman:** A POST request to `http://localhost:5000/auth_06/register_06` with a JSON body:

```
{
  "name": "Tony_zhan",
  "email": "abcdqazzaq@gmail.com",
  "password": "secret",
  "lastName": "lastName",
  "location": "my cityx",
  "_id": "62a62e6bcf7d636962d4694",
  "v": 0
}
```

The response status is 201 Created.
- MongoDB Compass:** Shows the `tonychickenDB_06.user_06` collection. The document displayed is:

```
{
  "_id": ObjectId('62a62e6bcf7d636962d4694'),
  "name": "Tony_zhan",
  "email": "abcdqazzaq@gmail.com",
  "password": "secret",
  "lastName": "lastName",
  "location": "my cityx",
  "v": 0
}
```

## Middelware Error page-code 500



83、84 gersalt for random AND JWT.sign for encode that work User secret (use Postman test this API)

The screenshot displays a Visual Studio Code editor with a project named '1102-WP2-MERN-207410506'. The Explorer sidebar on the left shows the file structure, including 'models', 'controllers', 'routes', and 'server\_06.js'. The main editor area shows the 'User\_06.js' file in the 'models' directory, which defines a Mongoose schema and a JWT creation method.

```
models > JS User_06.js > createJWT
42
43 UserSchema_06.pre('save', async function () {
44   // console.log('password',this.password);
45   const salt = await bcrypt.genSalt(10); //Salt產生亂碼，預設跑10次(足夠難破解)
46   this.password = await bcrypt.hash(this.
47     password, salt)
48 });
49
50 UserSchema_06.methods.createJWT=function(){
51   console.log("this",this); //this這邊表示輸入的全部的資料
52   return Jwt.sign({userId:this._id}, 'jwtsecret', {expiresIn: '1d'}); //jwt.sign 可創token
53 };
54
55 export default mongoose.model('User_06', UserSchema_06);
```

The PROBLEMS panel at the bottom shows a warning for an undefined property 'password' in the 'this' object. The TERMINAL panel shows the command 'pwsh' and 'bash' being executed.

The second screenshot shows the 'authController\_06.js' file in the 'controllers' directory, which handles the registration of a new user and the generation of a JWT token.

```
controllers > JS authController_06.js > register_06 > token
1 import User_06 from "../models/User_06.js";
2
3 const register_06 = async (req, res, next) => { //next 往後傳送
4   try {
5     console.log('body', req.body);
6     const user = await User_06.create(req.body); //寫進mongodb
7     const token=user.createJWT(); //使用model的方法createJWT
8
9     res.status(201).json({ user,token });
10    // res.send('register user -- Tony Zhan , 207410506');
11  } catch (err) {
12    // res.status(500).json({ msg: "Error on registering user" });
13  }
14 }
```

The Postman window on the right shows a POST request to 'http://localhost:3000/auth\_06/register\_06'. The response is a JSON object containing the user details and a JWT token.

```
1 {
2   "user": {
3     "name": "Tony_zhan",
4     "email": "abcdqazqzda@gmail.com",
5     "password": "$2a$10$3gnmoWjfw21RwyJo./VsXuAv0dNm1bXf",
6     "lastName": "lastName",
7     "location": "my city",
8     "_id": "62a6ad99c3f97133b5d7deec",
9     "__v": 0
10  },
11  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1aW50IjI2MmE2YWQ5OWMzZjk3MTMzYjVKN2R1ZWMiLCJpbnRknF6m1jImwAawPREyFkM4LABaoHRcg1oDr613X5M"
12 }
```

git log 紀錄

The screenshot shows the Visual Studio Code editor interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The editor has several tabs open: JS server\_06.js, JS authController\_06.js, .env, JS User\_06.js (active), JS error.handler\_06.js, JS Rigsiter\_06.js 6, and JS authRoutes\_06.js. The active file, User\_06.js, contains the following code:

```
models > JS User_06.js > ...
47   password, salt)
48   });
49
50   UserSchema_06.methods.createJWT=function(){
51     console.log("this",this);//this這邊表示輸入的全部的資料
52     return Jwt.sign({userId:this._id},process.env.JWT_SECRET,{expiresIn:process.env.JWT_LIFETIME});//jwt.sign 可創token
53   };
54
55   export default mongoose.model('User_06', UserSchema_06);
```

The bottom panel shows the TERMINAL view with the following output:

```
shinda@LAPTOP-DQ55BUTU MINGW64 ~/Desktop/週三週四網頁設計/網頁設計/1102-wp2-mern-207410506 (main)
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2022-06-12"
2b1dace tony.zhan    Mon Jun 13 11:25:14 2022 +0800    jsonwebtoken_Jwt_sign_to_encode_password_token
47219d8 tony.zhan    Mon Jun 13 04:37:46 2022 +0800    mongoose_pre_to_secret_password_use_hash_and_gensalt
8a08717 tony.zhan    Mon Jun 13 04:16:44 2022 +0800    npm_validator_isemail_error_msg_implement_with_next
915d0db tony.zhan    Mon Jun 13 02:31:18 2022 +0800    MongooseSchema_Model_to_Mongodb_And_await_req_from_postman_and_response
24c0ea8 tony.zhan    Mon Jun 13 01:11:28 2022 +0800    Middleware_appuse_expressJson
dcfca78 tony.zhan    Mon Jun 13 01:07:55 2022 +0800    MongoDB_appuse_Route_authController

shinda@LAPTOP-DQ55BUTU MINGW64 ~/Desktop/週三週四網頁設計/網頁設計/1102-wp2-mern-207410506 (main)
$
```

The status bar at the bottom shows the current file is 'main', the editor is in 'Scanning..' mode, and the Git Graph is visible. The bottom right corner of the window shows the system tray with the date and time: 上午 11:26, 2022/6/13.