

24.01.15

🕒 작성 일시	@2024년 1월 15일 오전 10:14
🏷 태그	

```
powerShell 관리자 권한 변경

PS C:\work\vue-axios\backend> Set-ExecutionPolicy RemoteSigned

실행 규칙 변경
실행 정책은 신뢰하지 않는 스크립트로부터 사용자를 보호합니다. 실행 정책을 변경하면 about_Execution_Policies 도움말
항목(https://go.microsoft.com/fwlink/?LinkID=135170)에 설명된 보안 위험에 노출될 수 있습니다. 실행 정책을
변경하시겠습니까?
[Y] 예(Y) [A] 모두 예(A) [N] 아니요(N) [L] 모두 아니요(L) [S] 일시 중단(S) [?] 도움말 (기본값은 "N"): y
PS C:\work\vue-axios\backend>

npm init

PS C:\windows\system32> cd c:\work
PS C:\work> md vue-axios
PS C:\work> cd vue-axios
PS C:\work\vue-axios> md backend
PS C:\work\vue-axios\backend> npm init -y
Wrote to C:\work\vue-axios\backend\package.json:

{
  "name": "backend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

PS C:\work\vue-axios\backend> npm install express mysql2 cors
added 75 packages, and audited 76 packages in 3s

11 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

npm install express mysql2 cors

EXPLORER

- ✓ VUE-AXIOS
 - ✓ backend
 - ✓ config
 - JS database.js
 - > node_modules
 - { } package-lock.json
 - { } package.json

backend > { } package.json > ...

```
1 {
2   "name": "backend",
3   "version": "1.0.0",
4   "description": "",
5   "type": "module",
6   "main": "index.js",
7   "scripts": {
8     "test": "echo \"Error: no test specified\" && exit 1"
9   },
10  "keywords": [],
11  "author": "",
12  "license": "ISC",
13  "dependencies": {
14    "cors": "^2.8.5",
15    "express": "^4.18.2",
16    "mysql2": "^3.7.0"
17  }
18 }
19
```

추가로 작성

Router 기본 설정

EXPLORER

- ✓ VUE-AXIOS
 - ✓ backend
 - ✓ config
 - JS database.js
 - > node_modules
 - ✓ router
 - ✓ controllers
 - JS Product.js
 - JS router.js
 - { } package-lock.json
 - { } package.json

backend > router > JS router.js > ...

```
1 import express from "express";
2
3 const router = express.Router();
```

백엔드 기본구성

✓ VUE-AXIOS

✓ backend

✓ config

JS database.js

> node_modules

✓ router

✓ controllers

JS Product.js

✓ Models

JS productModel.js

JS router.js

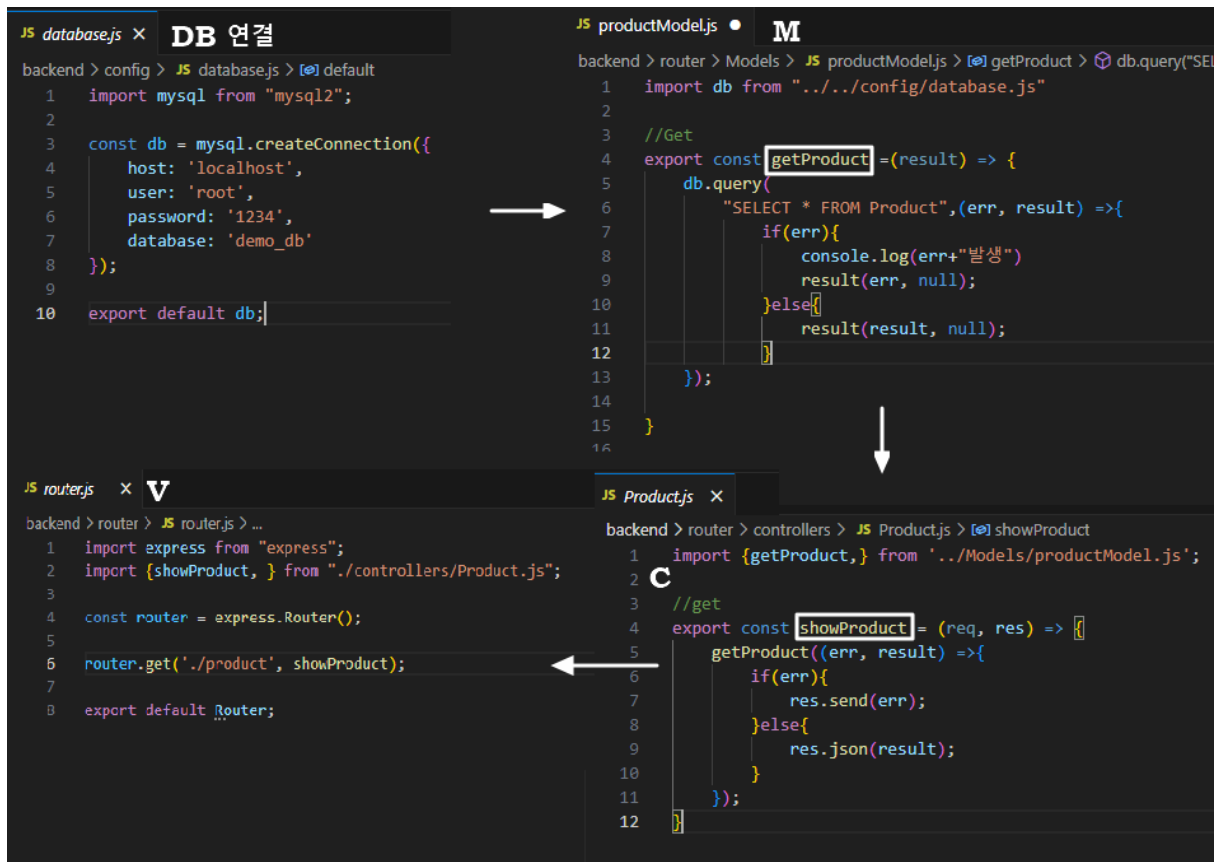
JS index.js

{ } package-lock.json

{ } package.json

The screenshot shows the VS Code interface with the Explorer sidebar on the left displaying the project structure. The main editor area shows the content of `index.js`.

```
backend > JS index.js > ...
1  import Express from "express";
2  import cors from "cors";
3  import Router from "../router/router.js";
4
5  app.listen(5000, () => console.log('Server running at http://localhost:5000')); //DB 와 백엔드(express)서버의 통신 포트 번호
```



백엔드 프론트 연결

```
PS C:\work\vue-axios2> cd frontend
```

```
PS C:\work\vue-axios2\frontend> npm install vue-router axios bulma
```

axios : 백엔드 서버의 내용을 뿌려주는 역할

npm install vue-router axios bulma

```

1 // import connection
2 import db from "../../config/database.js";
3
4 // Get All Products
5 export const getProducts = (result) => {
6   db.query("SELECT * FROM product", (err, results) => {
7     if(err) {
8       console.log(err);
9       result(err, null);
10    } else {
11      result(null, results);
12    }
13  });
14 }
15
16 // Get Single Product      id 기준으로 탐색
17 export const getProductById = (id, result) => {
18   db.query("SELECT * FROM product WHERE product_id = ?", [id], (err, results) => {
19     if(err) {
20       console.log(err);
21       result(err, null);
22     } else {
23       result(null, results[0]);
24     }
25   });
26 }
27
28 // Insert Product to Database
29 export const insertProduct = (data, result) => {
30   db.query("INSERT INTO product SET ?", [data], (err, results) => {
31     if(err) {
32       console.log(err);
33       result(err, null);
34     } else {
35       result(null, results);
36     }
37   });
38 }
39
40 // Update Product to Database      id 기준으로 탐색
41 export const updateProductById = (data, id, result) => {
42   db.query("UPDATE product SET product_name = ?, product_price = ? WHERE product_id = ?", [data.product_name, data.product_price, id], (err, results) => {
43     if(err) {
44       console.log(err);
45       result(err, null);
46     } else {
47       result(null, results);
48     }
49   });
50 }
51
52 // Delete Product to Database      id 기준으로 탐색
53 export const deleteProductById = (id, result) => {
54   db.query("DELETE FROM product WHERE product_id = ?", [id], (err, results) => {
55     if(err) {
56       console.log(err);
57       result(err, null);
58     } else {
59       result(null, results);
60     }
61   });
62 }

```