

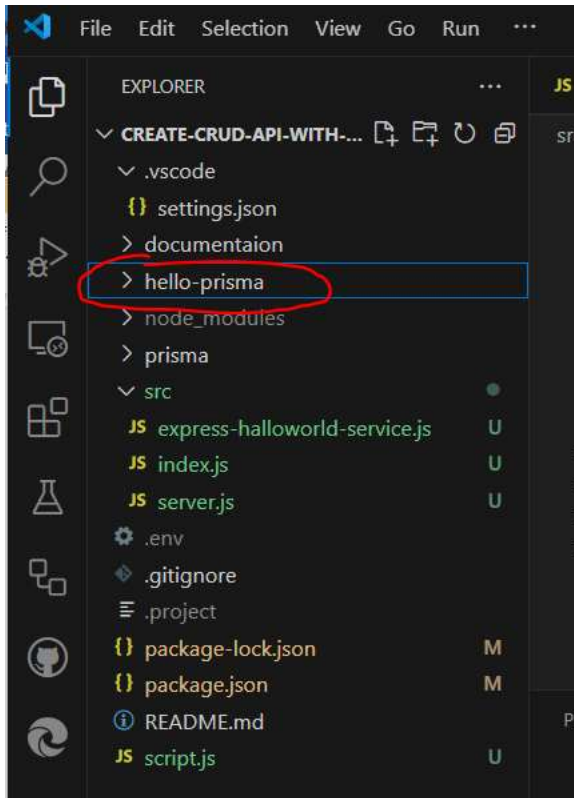
This app is based on my “hello-prisma” basic prisma app

<https://github.com/CHDPrismaApps/hello-prisma>

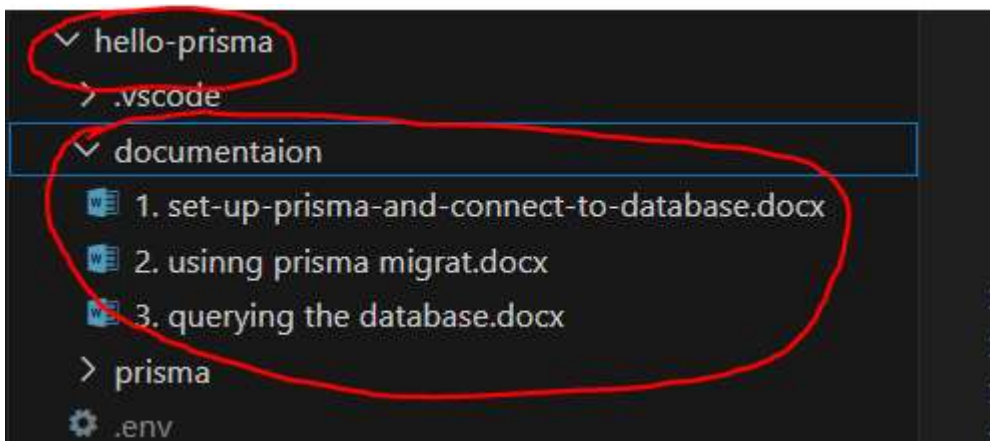
➔ Copy “hello-prisma” from my GitHub to here.

<https://github.com/CHDPrismaApps/hello-prisma>

The prisma basic app can also be found in my sub-folder: **hello-prisma**.

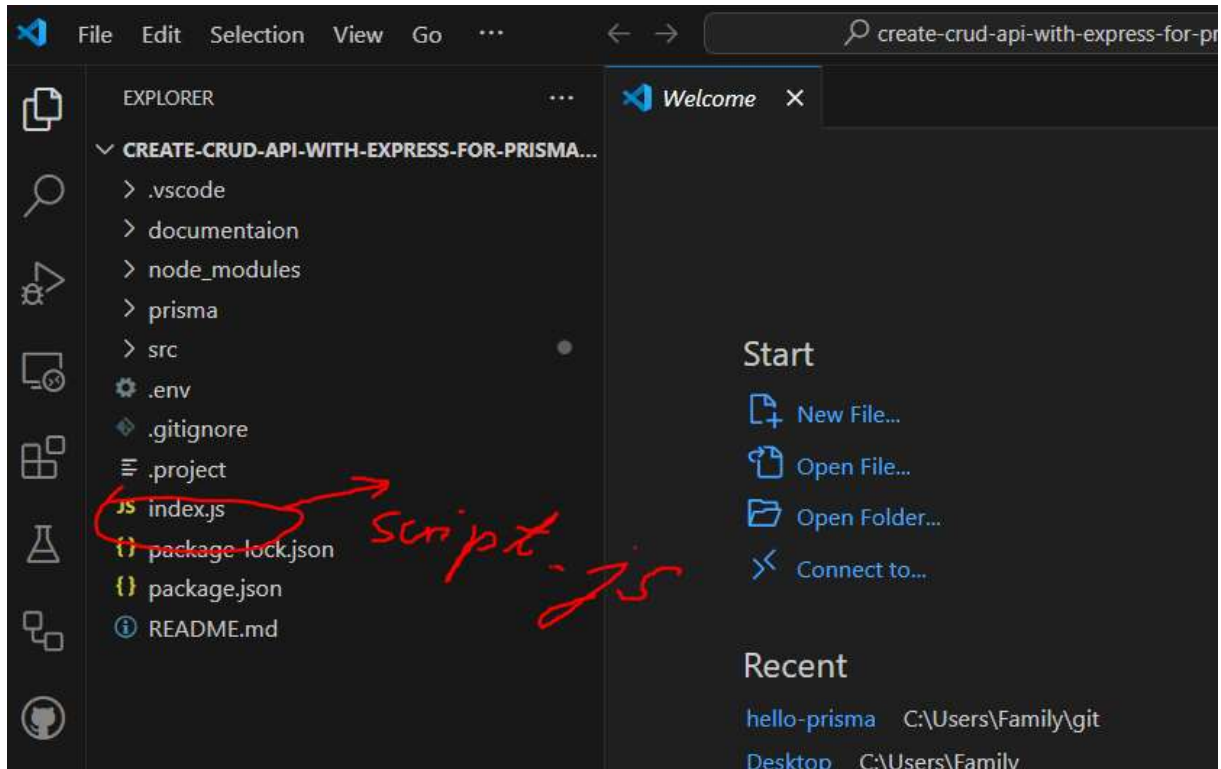


Please read the following 3 documents of **hello-prisma** first to understand where the codes come from.

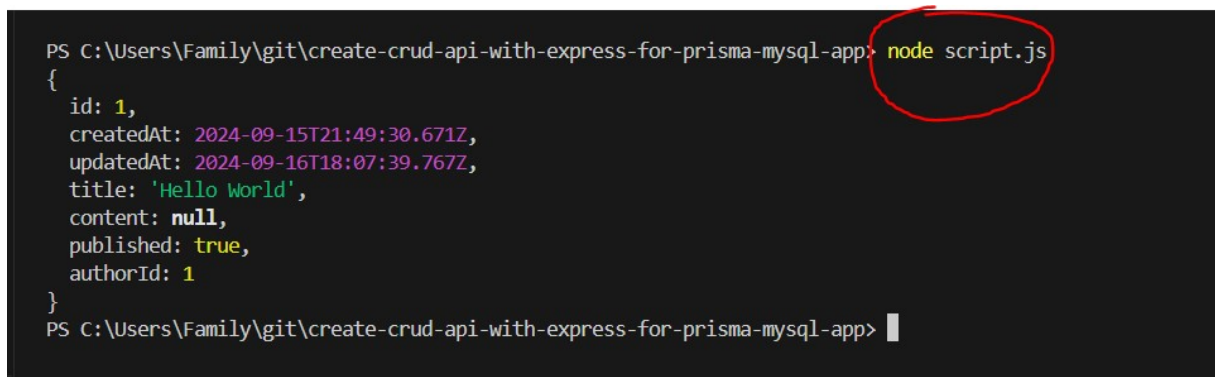


## ➔ Refactoring

Rename ➔ index.js to script.js



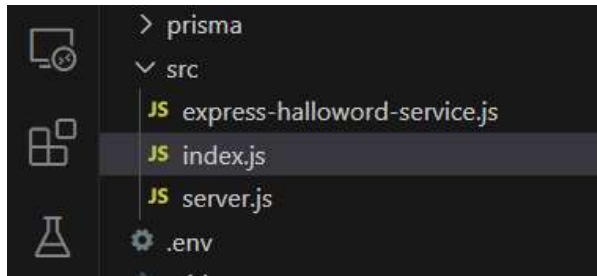
C:\Users\Family\git\hello-prisma> node script.js



RUN... ➔ It works – ok

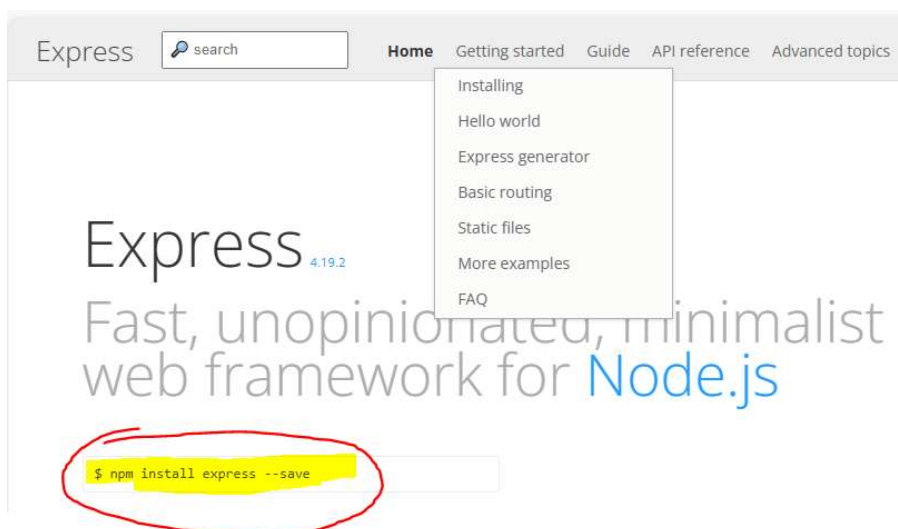
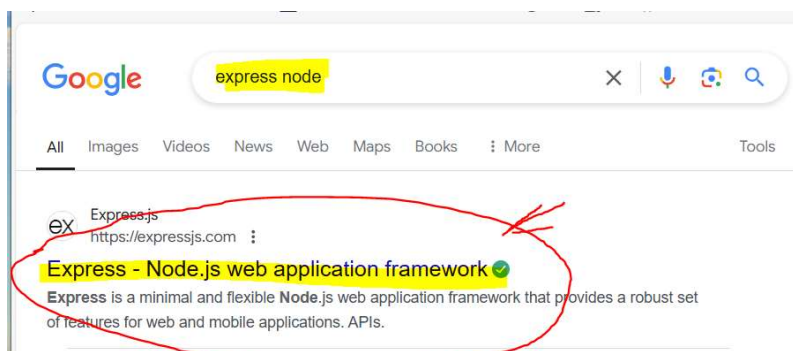
Create a **src** folder and add **three files**:

- **express-halloword-service.js**
- **index.js**
- **server.js**



➔ To create Express Server and Endpoints.

We need first to **install express** (<https://expressjs.com/>).



`npm install express --save`

➔ Install **express**

C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app> **npm**  
**install express --save**

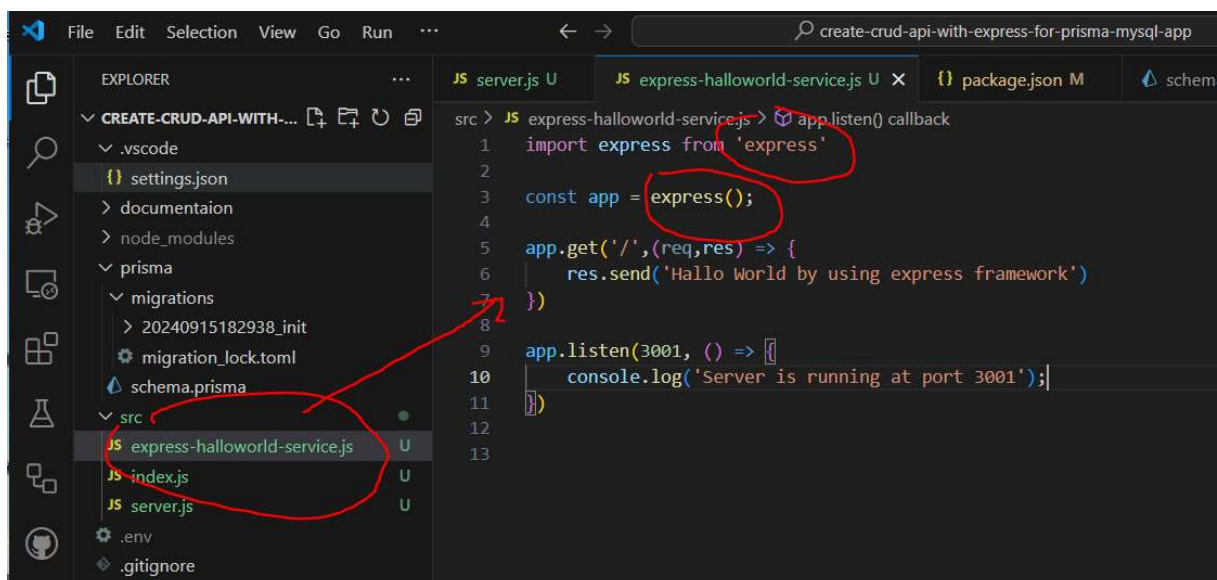
```
PS C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app> npm install express --save
added 65 packages, and audited 93 packages in 3s

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

found 0 vulnerabilities
PS C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app>
```

➔ **express** was installed.

Edit testcase to **express-helloworld-service.js**.



**express-helloworld-service.js**

```
import express from 'express'
const app = express();
app.get('/', (req, res) => {
  res.send('Hallo World by using express framework: https://expressjs.com/')
})
app.listen(3001, () => {
  console.log('Server is running at port 3001');
})
```

➔ RUN...

```
PS C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app> node ./express-halloword-service.js
```

```
C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app>  
node ./express-halloword-service.js
```

```
Error: Cannot find module 'C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app\express-halloword-service.js'  
    at Function.Module._resolveFilename (node:internal/modules/cjs/loader:985:15)  
    at Function.Module._load (node:internal/modules/cjs/loader:833:27)  
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:81:12)  
    at node:internal/main/run_main_module:22:47 {  
  code: 'MODULE_NOT_FOUND',  
  requireStack: []  
}
```

Got **ERROR**. Install the following dependencies.

```
npm install --save @types/node  
npm install --save @types/express
```

```
C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app> npm  
install --save @types/node
```

```
PS C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app> npm install --save @types/node  
  
up to date, audited 93 packages in 748ms  
  
13 packages are looking for funding  
  run `npm fund` for details  
  
found 0 vulnerabilities  
PS C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app>
```

➔ Installed OK

```
C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app> npm  
install --save @types/express
```

```
PS C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app> npm install --save @types/express  
  
added 10 packages, and audited 103 packages in 1s  
  
13 packages are looking for funding  
  run `npm fund` for details  
  
found 0 vulnerabilities  
PS C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app>
```

➔ Installed OK



```
C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app> node  
./express-helloworld-service.js
```

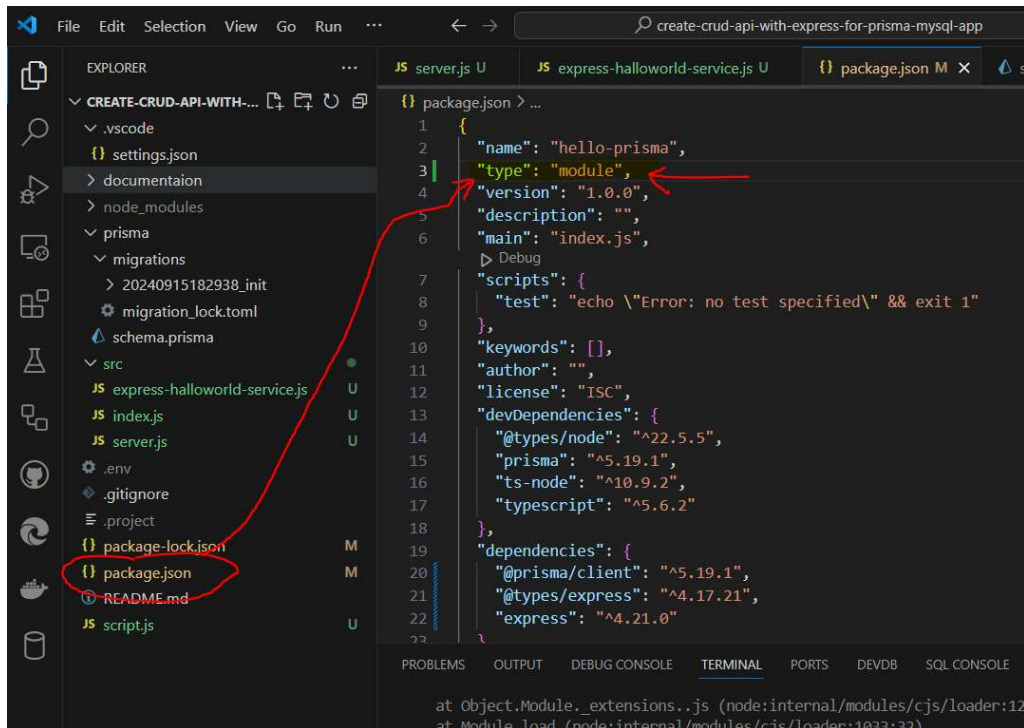
Got **ERROR**.

```
PS C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app> node src/express-helloworld-service.js  
(node:22648) Warning: To load an ES module, set "type": "module" in the package.json or use the .mjs extension.  
(Use "node --trace-warnings ..." to show where the warning was created)  
C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app\src\express-helloworld-service.js:1  
import express from './express';  
SyntaxError: Cannot use import statement outside a module  
    at Object.compileFunction (node:vm:360:18)  
    at wrapSafe (node:internal/modules/cjs/loader:1084:15)  
    at Module._compile (node:internal/modules/cjs/loader:1119:27)  
    at Object.Module._extensions..js (node:internal/modules/cjs/loader:1209:10)  
    at Module.load (node:internal/modules/cjs/loader:1033:32)  
    at Function.Module._load (node:internal/modules/cjs/loader:868:12)  
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:81:12)  
    at node:internal/main/run_main_module:22:47
```

```
SyntaxError: Cannot use import statement outside a module  
    at Object.compileFunction (node:vm:360:18)  
    at wrapSafe (node:internal/modules/cjs/loader:1084:15)  
    at Module._compile (node:internal/modules/cjs/loader:1119:27)  
    at Object.Module._extensions..js (node:internal/modules/cjs/loader:1209:10)  
    at Module.load (node:internal/modules/cjs/loader:1033:32)  
    at Function.Module._load (node:internal/modules/cjs/loader:868:12)  
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:81:12)  
    at node:internal/main/run_main_module:22:47
```



On package.json → add "type": "module"

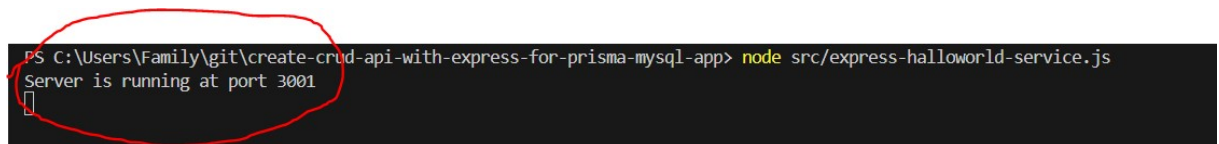


The screenshot shows the Visual Studio Code interface. In the Explorer on the left, the file `package.json` is selected and circled in red. The main editor displays the contents of `package.json`. A red arrow points to the line `"type": "module",` which has been added to the `package.json` object. The file is named `hello-prisma` and has version `1.0.0`. The `main` field is `index.js`. The `scripts` field includes a `test` script. The `devDependencies` field lists `@types/node`, `prisma`, `ts-node`, and `typescript`. The `dependencies` field lists `@prisma/client`, `@types/express`, and `express`.

```
1 {
2   "name": "hello-prisma",
3   "type": "module",
4   "version": "1.0.0",
5   "description": "",
6   "main": "index.js",
7   "scripts": {
8     "test": "echo \\\"Error: no test specified\\\" && exit 1"
9   },
10  "keywords": [],
11  "author": "",
12  "license": "ISC",
13  "devDependencies": {
14    "@types/node": "^22.5.5",
15    "prisma": "^5.19.1",
16    "ts-node": "^10.9.2",
17    "typescript": "^5.6.2"
18  },
19  "dependencies": {
20    "@prisma/client": "^5.19.1",
21    "@types/express": "^4.17.21",
22    "express": "^4.21.0"
23  }
24 }
```

➔ RUN...

C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app>  
node src/express-helloworld-service.js



The screenshot shows a terminal window with the command `node src/express-helloworld-service.js` executed. The output is `Server is running at port 3001`, which is circled in red.

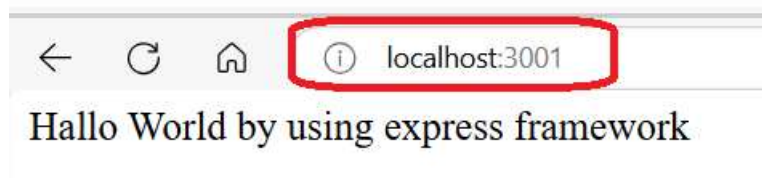
```
PS C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app> node src/express-helloworld-service.js
Server is running at port 3001
```

➔ Server is running at port 3001

Review the code: `res.send('Hallo World by using express framework')`

```
src > 5 express-halloworld-service.js > ..  
1  import express from 'express'  
2  
3  const app = express();  
4  
5  app.get('/', (req, res) => {  
6    res.send('Hallo World by using express framework')  
7  })  
8  
9  app.listen(3001, () => {  
10    console.log('Server is running at port 3001');  
11  })  
12  
13
```

Open browser and type localhost:3001

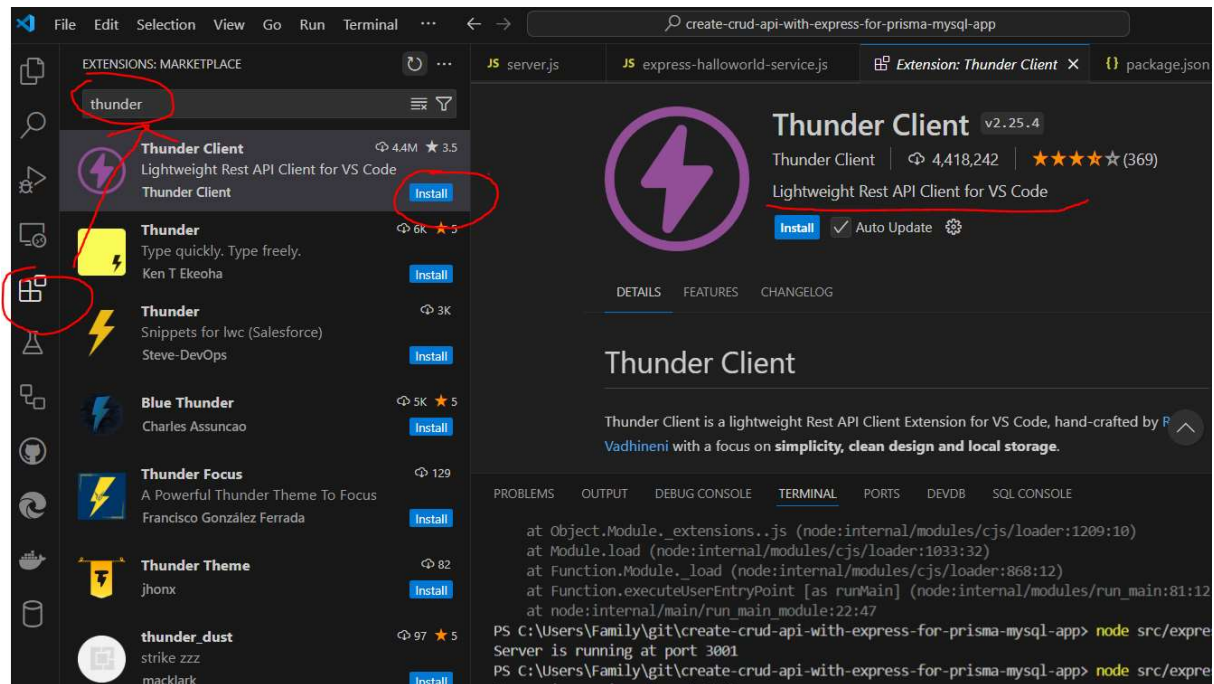


The screenshot shows a web browser window. The address bar contains the URL 'localhost:3001', which is highlighted with a red rectangle. Below the address bar, the text 'Hallo World by using express framework' is displayed.

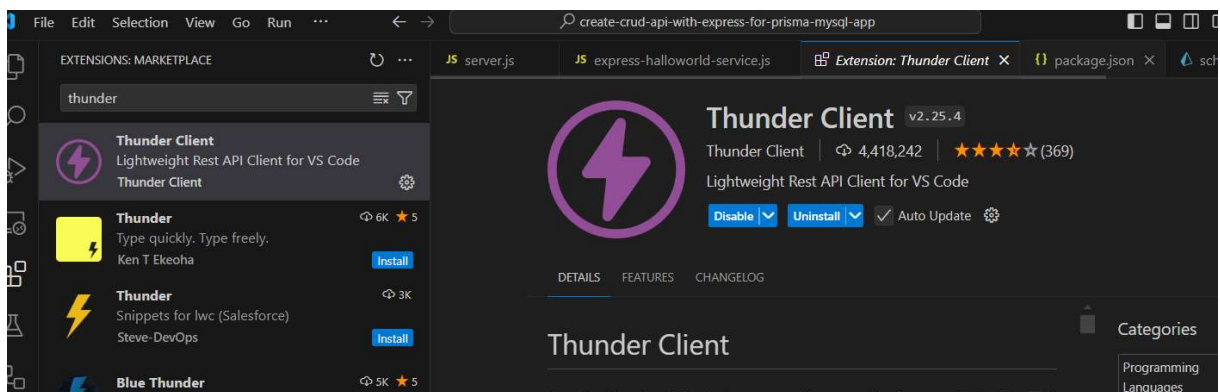
➔ Successful. OK 😊



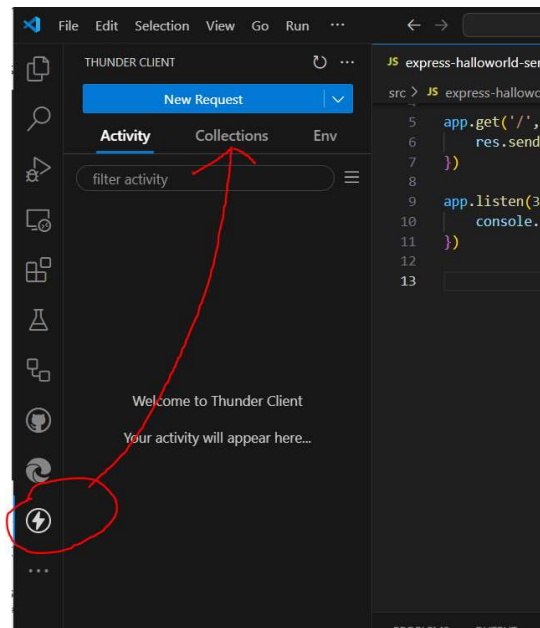
Extensions → Install **http client** by using Thunder Client  
→ **Rest API Client for VS Code**



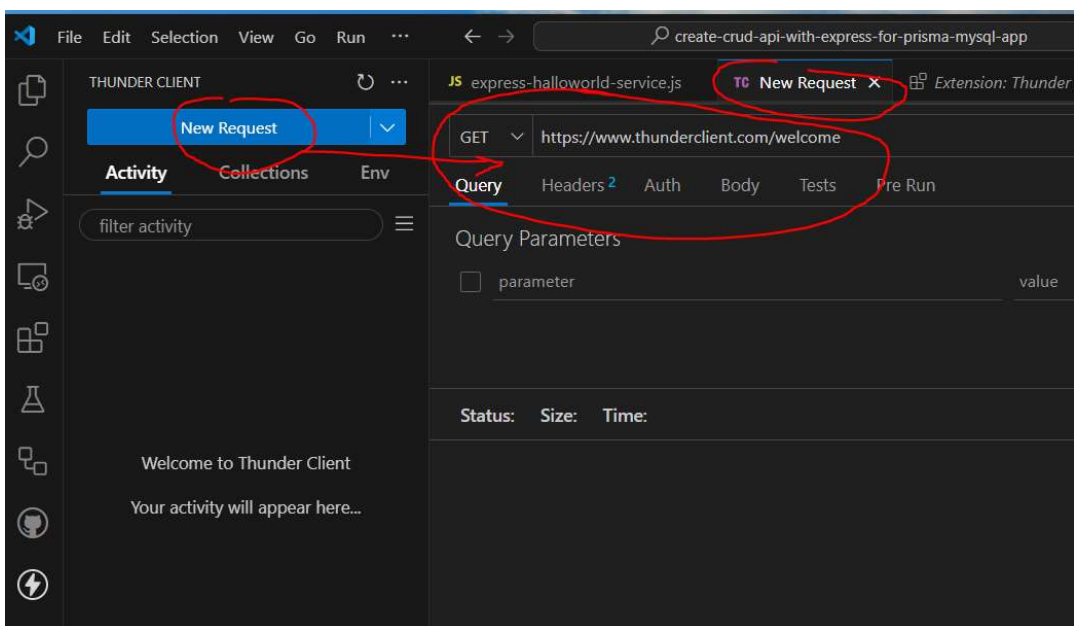
Thunder Client was installed. → OK



Click → Thunder Client



We got the default **GET** request as below:

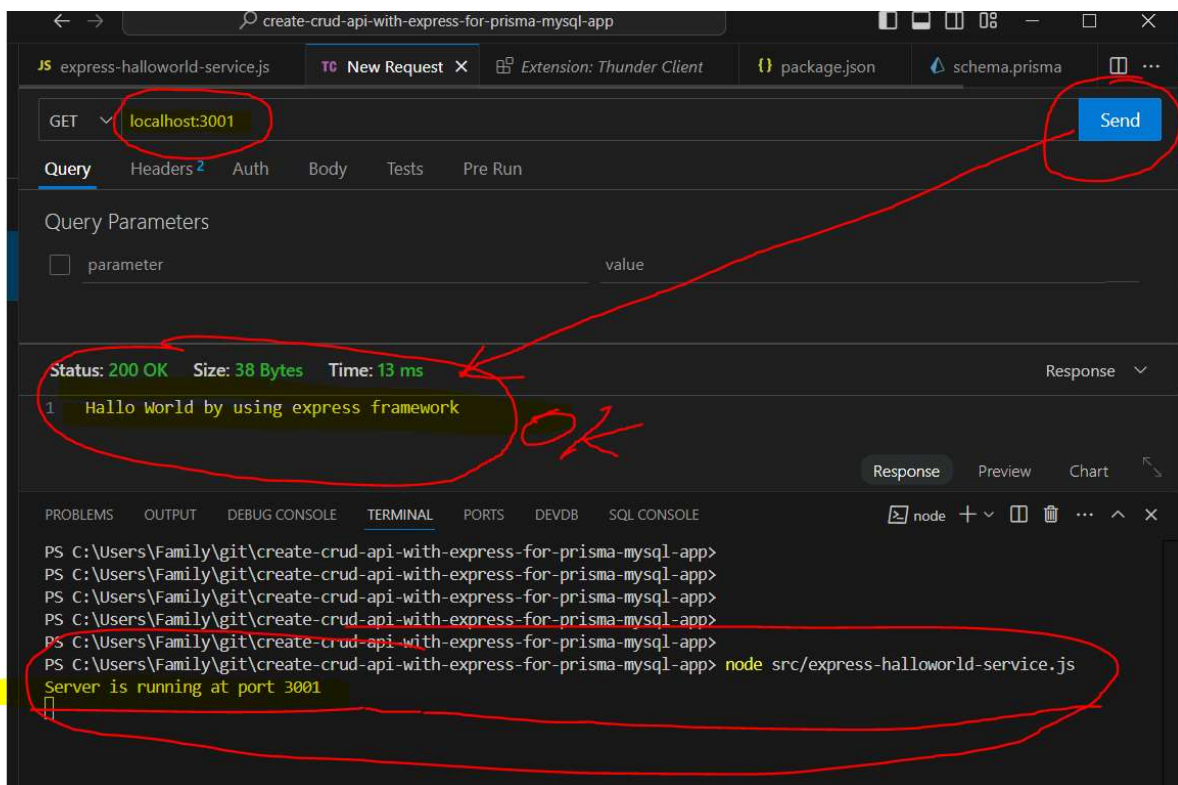


➔ RUN...

C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app> node  
./express-helloworld-service.js

```
PS C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app> node src/express-helloworld-service.js  
Server is running at port 3001
```

Change the GET request to localhost:3001 and click Send button → Response Status = 200 OK → The result is exactly what we want: “Hallo World by using express framework”



Rest API Client | http Client → Successful 😊