

How to implement a simple restful API using Express and React

井民全, Jing, mqijing@gmail.com

Here, I'll show you how to create a simple restfulAPI in few steps.

Enjoy!

By Jing.

1. Create the basic Express/React skeleton

1.1. Tree

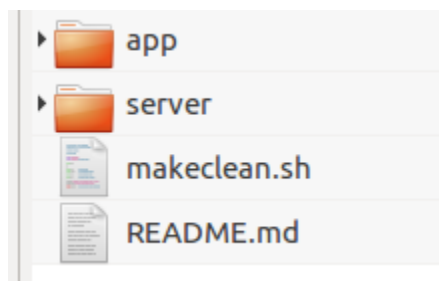


Fig. The tree structure.

1.2. The skeleton

```
# init: create tree structure
```

```
# Step 1: create react app
```

```
npx create-react-app app --template typescript
```

```
cd app
yarn build
cd ../../

# Step 2: create express server
mkdir server
cd server
tsc --init
yarn init
yarn add express @types/express nodemon
```

```
# Step 3: write service code
File: index.ts (app/server/index.ts)
```

```
import express from 'express';
import path from 'path';

const app = express();
app.use(express.static(path.join(__dirname, '../app/build')));
app.get('/', function (req, res) {
  res.sendFile(path.join(__dirname, '../app/build/index.html'));
});

app.listen(9000);
```

```
# Step 4: run
tsc
node index.js
```

```
# Test
gio open http://localhost:9000
```

2. Create router to Server

2.1. new router

Step 1: Create router folder

```
mkdir routes
```

Step 2: Create a router

File: server/routes/statusAPI.ts

```
import express, {Request, Response} from 'express';

var router = express.Router();

router.get('/', function(req:Request, res:Response, next) {
  res.send('statusAPI invoked');
});

module.exports = router;
```

2.2. Insert router

Step 1: Require the module

```
var statusAPIRouter = require('./routes/statusAPI');
```

Step 2: Insert it

```
app.use('/statusAPI', statusAPIRouter);
```

File: server/index.ts

```
import express from 'express';
import path from 'path';

var statusAPIRouter = require('./routes/statusAPI');
const app = express();

app.use(express.static(path.join(__dirname, '../app/build'))); // <-- this the react build folder
app.use('/statusAPI', statusAPIRouter);

app.get('/', function (req, res) {
  res.sendFile(path.join(__dirname, '../app/build/index.html'));
});

app.listen(9000);
```

3. Build & Test & Clean

3.1. Build

```
# Express server part
cd server
yarn install
tsc
node index.js

# React part
cd app
yarn install
yarn build
```

3.2. Test

```
gio http://localhost:9000/statusAPI'
```

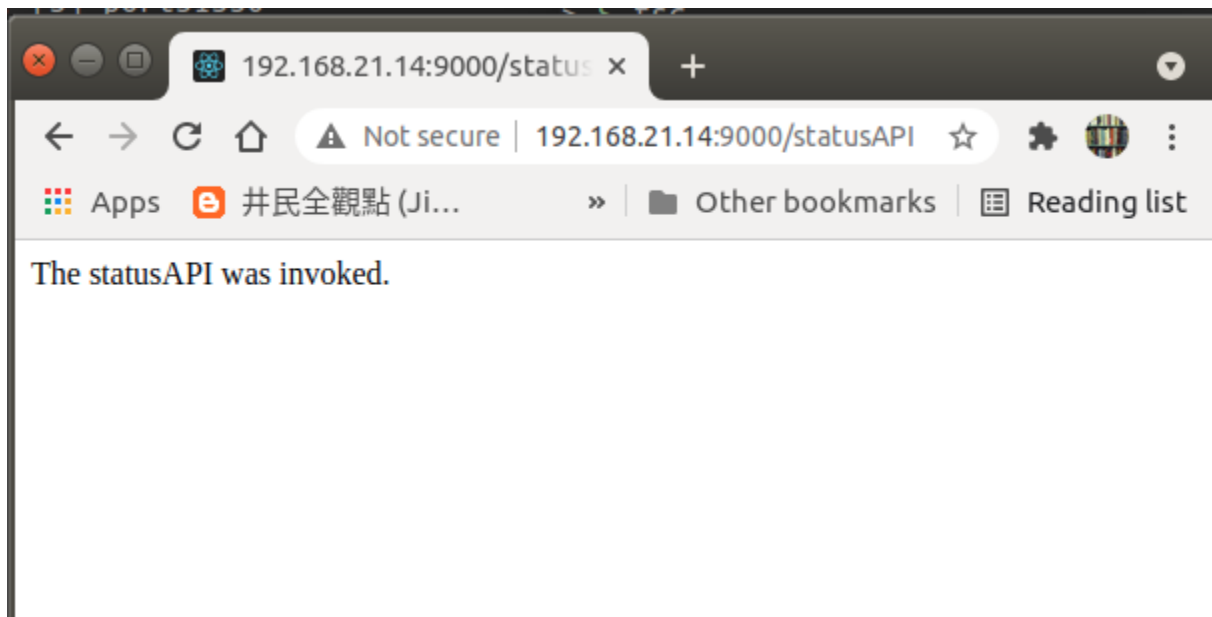


Fig. Test statusAPI.

3.3. Clean

File: server/**package**.json and app/**package**.json

```
...  
  "scripts": {  
    ...  
    "clean": "(rm -fr node_modules; rm -fr build; find . -name \"*.js\" -type f|xargs rm -f; find .  
-name \"*.map\" -type f|xargs rm -f)"  
    ...  
  },  
  ...  
}
```

4. Reference

1. <https://www.freecodecamp.org/news/create-a-react-frontend-a-node-express-backend-and-connect-them-together-c5798926047c/>
- 2.