

CT449: PHÁT TRIỂN ỨNG DỤNG WEB

Ứng dụng Contactbook - Backend - Phần 1

Bước 0: Cài đặt node và git

- Tải và cài đặt nodejs: <https://nodejs.org/en/download/>. Có thể tải và cài nodejs trực tiếp hoặc cài nodejs thông qua nvm. (<https://github.com/coreybutler/nvm-windows>).
- Tải và cài đặt git tại: <https://git-scm.com/download/win>.
- Để kiểm tra rằng có thể gõ lệnh node và git trên Terminal / Command Prompt:

```
E:\HK2\ud_web\ngtrgiabao>node -v
v16.14.2

E:\HK2\ud_web\ngtrgiabao>git --version
git version 2.37.3.windows.1

E:\HK2\ud_web\ngtrgiabao>_
```

Bước 1: Tạo ứng dụng Node

- Tạo thư mục dự án, ví dụ "*contactbook-backend*".
- Trong thư mục này, ta khởi tạo ứng dụng Node:

```
npm init
```

`npm` sẽ hỏi một số thông tin để tạo tập tin package.json cho dự án:

```

package name: (contactbook-backend)
version: (1.0.0)
description:
entry point: (index.js) server.js
test command:
git repository:
keywords:
author: Gia Bao
license: (ISC)
About to write to E:\HK2\ud_web\ngtrgiabao\contactbook-backend\package.json:

{
  "name": "contactbook-backend",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Gia Bao",
  "license": "ISC"
}

Is this OK? (yes) █

```

- Cài đặt các package cần thiết:

```
npm install express cors
```

Bước 2: Quản lý mã nguồn dự án với git và GitHub

- Tạo tập tin .gitignore: npx gitignore node . Tập tin .gitignore liệt kê các thư mục, tập tin sẽ không được quản lý bởi git.
- Trong thư mục gốc dự án, chạy lệnh git init. Gõ tiếp git status, ta sẽ thấy có 3 tập tin không được quản lý bởi git là .gitignore, package-lock.json và package.json.

```

PS E:\HK2\ud_web\ngtrgiabao\contactbook-backend> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        package-lock.json
        package.json

nothing added to commit but untracked files present (use "git add" to track)

```

- Ta tiến hành cho git quản lý 3 tập tin .gitignore, package-lock.json và package.json bằng lệnh sau đây: git add .
- Gõ git status để kiểm tra lại

```
PS E:\HK2\ud_web\ngtrgiabao\contactbook-backend> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   package-lock.json
        new file:   package.json
```

- Thực thi `git commit -m "Cai dat du an"` để lưu các thay đổi. Tùy chọn `-m` trong câu lệnh cho phép ta tạo 1 ghi chú cho commit này.
- Đăng ký tài khoản và tạo dự án trên GitHub. Sau khi tạo, chúng ta upload dự án cục bộ lên GitHub như sau:

```
git remote add origin <github-url>
git push origin master
```

- Trong đó, ta sẽ thay thế `<github-url>` bằng URL của dự án đã tạo trên GitHub.
- Thực hiện chứng thực tài khoản và kiểm tra lại các tập tin đã được tải lên GitHub:

```
PS E:\HK2\ud_web\ngtrgiabao\contactbook-backend> git remote add origin https://github.com/ngtrgiabaoB2012063/contactbook-backend.git
PS E:\HK2\ud_web\ngtrgiabao\contactbook-backend> git push -u origin master
```

```
nothing to commit, working tree clean
PS E:\HK2\ud_web\ngtrgiabao\contactbook-backend> git commit -m "cai dat du an"
On branch master
nothing to commit, working tree clean
```

master
1 branch
0 tags
Go to file
Add file
Code

ngtrgiabaoB2012063 cai dat du an
f2ab294 3 minutes ago
1 commit

	.gitignore	cai dat du an	3 minutes ago
	package-lock.json	cai dat du an	3 minutes ago
	package.json	cai dat du an	3 minutes ago

Bước 3: Cài đặt Express vào chương trình

Trong thư mục gốc của dự án, tạo tập tin `app.js` và `server.js` lần lượt với nội dung sau:

- Tập tin `app.js`



```
const express = require("express");
const cors = require("cors");

const app = express();
const contactsRouter = require(
  "./app/routes/contact.route");

app.use(express.json());
app.use(cors());

app.get("/", (req, res) => {
  res.send({message:
    "Welcome to contact book application."});
});

module.exports = app;
```

- Tập tin server.js



```
const app = require("./app");
const config = require("./app/config");

const PORT = config.app.port;

app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```

Lần lượt tạo thư mục *app*, *app/config* và tập tin *app/config/index.js*:

```
const config = {
  app: {
    port: process.env.PORT || 3000,
  },
};
module.exports = config;
```

Mở cửa sổ terminal tại thư mục gốc của dự án và thực hiện lệnh chạy server: node server.js

Truy cập <http://localhost:3000/> để kiểm tra kết quả đã thành công chưa.

Để dễ dàng hơn cho việc phát triển code sau này, chúng ta có thể cài đặt thêm gói hỗ trợ nodemon để kiểm soát việc thay đổi code tốt hơn.

Gói nodemon này sẽ giám sát các thay đổi trên mã nguồn và tự động khởi động server:

```
PS E:\HK2\ud_web\ngtrgiabao\contactbook-backend> npm i nodemon
```

Sau khi cài đặt, mở tập tin package.json và thay đổi cấu hình mục "scripts" như sau:

```
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1",
  "start": "nodemon server.js"
},
```

Với cấu hình như trên, chúng ta có thể chạy server bằng lệnh npm start thay vì node server.js và mỗi lần các tập tin bị thay đổi, nodemon sẽ tự động khởi động lại server giúp chúng ta.

Lưu các thay đổi trên vào git:

```
PS E:\HK2\ud_web\ngtrgiabao\contactbook-backend> git add -u
warning: in the working copy of 'package-lock.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'package.json', LF will be replaced by CRLF the next time Git touches it
PS E:\HK2\ud_web\ngtrgiabao\contactbook-backend> git add .
PS E:\HK2\ud_web\ngtrgiabao\contactbook-backend> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   app.js
    new file:   app/config/index.js
    modified:   package-lock.json
    modified:   package.json
    new file:   server.js
PS E:\HK2\ud_web\ngtrgiabao\contactbook-backend> git commit -m "Cai dat express chay thong bao xin chao"
```

Bước 4: Định nghĩa Controller và các route trong chương trình

Trong thư mục app/controllers, tạo tập tin contact.controller.js với nội dung sau:

```
exports.create = (req, res) => {
  res.send({ msg: "create handle" });
};

exports.findAll = (req, res) => {
  res.send({ msg: "findAll handle" });
};

exports.findOne = (req, res) => {
  res.send({ msg: "findOne handle" });
};

exports.update = (req, res) => {
  res.send({ msg: "update handle" });
};

exports.delete = (req, res) => {
  res.send({ msg: "delete handle" });
};

exports.deleteAll = (req, res) => {
  res.send({ msg: "delete all user handle" });
};

exports.findAllFavorite = (req, res) => {
  res.send({ msg: "find favorite contact handle" });
};
```

Trong thư mục app/routes, tạo tập tin contact.route.js với nội dung sau:

```
const express = require("express");
const contacts = require("../controllers/app.controller");

const router = express.Router();

router
  .route("/")
  .get(contacts.findAll)
  .post(contacts.create)
  .delete(contacts.deleteAll);

router.route("/favorite").get(contacts.findAllFavorite);

router
  .route("/:id")
  .get(contacts.findOne)
  .put(contacts.update)
  .delete(contacts.delete);

module.exports = router;
```

Trong đoạn code trên, chúng ta định nghĩa các route quản lý liên hệ được hỗ trợ bởi ứng dụng web. Mỗi route là sự kết hợp của đường dẫn, phương thức HTTP (GET, POST, PUT, DELETE) và đoạn code xử lý.

Tiếp theo, chúng ta sẽ đăng ký các route vừa tạo với express app. Điều chỉnh tập tin app.js như sau:

```
const app = express();
const contactsRouter = require("./app/routes/contact.route");

app.use(express.json());
app.use(cors());
app.use("/api/contacts", contactsRouter);
```

Sau khi đảm bảo các route hoạt động tốt trên server, ta lưu các thay đổi lên git:

```
PS E:\HK2\ud_web\ngtrgiabao\contactbook-backend> git add -u
PS E:\HK2\ud_web\ngtrgiabao\contactbook-backend> git add .
PS E:\HK2\ud_web\ngtrgiabao\contactbook-backend> git commit -m "Định nghĩa các route cho tài nguyên Contact"
[master 40413f9] Định nghĩa các route cho tài nguyên Contact
3 files changed, 49 insertions(+), 1 deletion(-)
create mode 100644 app/controllers/app.controller.js
create mode 100644 app/routes/contact.route.js
```

Bước 5: Cài đặt xử lý lỗi trong chương trình

Tạo tập tin app/api-error.js:

```
class ApiError extends Error {
  constructor(statusCode, message) {
    super();
    this.statusCode = statusCode;
    this.message = message;
  }
}
```

Mở tập tin app.js và thêm vào các middleware xử lý lỗi như sau:

```
const express = require("express");
const cors = require("cors");

const app = express();
const contactsRouter = require("../app/routes/contact.route");

app.use(express.json());
app.use(cors());
app.use("/api/contacts", contactsRouter);

app.get("/", (req, res) => {
  res.send({ message: "Welcome to contact book application." });
});

app.use("/api/contacts", contactsRouter);
// handle 404 response
app.use((req, res, next) => {
  return next(new ApiError(404, "Resource not found"));
});

app.use((error, req, res, next) => {
  return res.status(error.statusCode || 500).json({
    message: error.message || "Internal Server Error",
  });
});

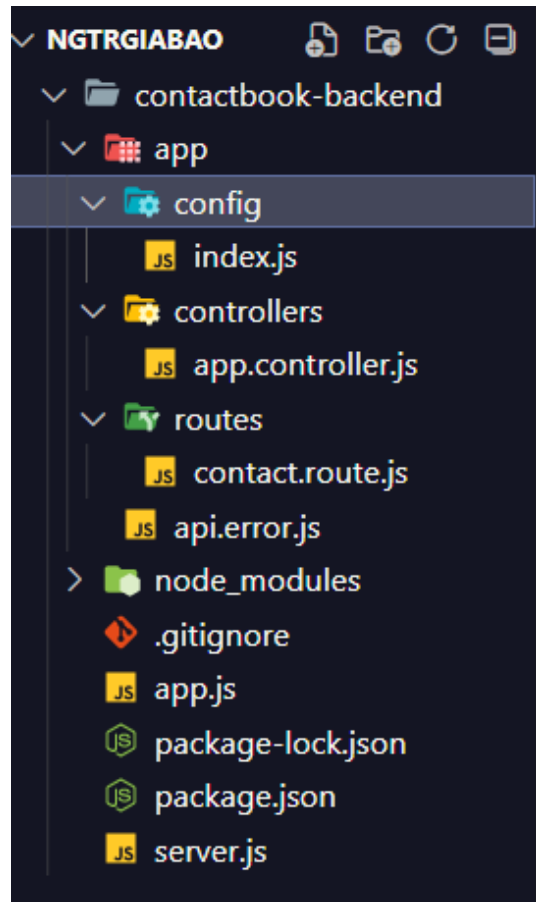
module.exports = app;
```

Sử dụng một HTTP client (Chrome, Postman, curl, ...) và gửi yêu cầu đến một URL không được định nghĩa trong ứng dụng, kiểm tra nhận được lỗi 404 và thông báo "Resource not found".

Lưu các thay đổi vào git:

```
PS E:\HK2\ud_web\ngtrgiabao\contactbook-backend> git add .
PS E:\HK2\ud_web\ngtrgiabao\contactbook-backend> git commit -m "Cai dat xu ly loi"
[master 240cedc] Cai dat xu ly loi
2 files changed, 19 insertions(+)
create mode 100644 app/api.error.js
```


Sau khi hoàn tất, ta tiến hành đẩy các thay đổi của dự án lên GitHub bằng lệnh:
git push origin master.



Link GitHub: <https://github.com/ngtrgiabaoB2012063/contactbook-backend.git>