

ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



PHÁT TRIỂN ỨNG DỤNG WEB
BACKEND

Giáo viên hướng dẫn:

Ths.GVC Nguyễn Minh Trung

Sinh viên thực hiện

Họ tên: Nguyễn Thanh Bình

MSSV: B2113327

Lớp học phần: CT44902

CẦN THƠ

Mục lục

Part 1	1
Bước 0: Cài đặt node và git.....	1
Bước 1: Tạo ứng dụng Node	1
Bước 2: Quản lý mã nguồn dự án với git và GitHub	3
Bước 3: Cài đặt Express	4
Bước 4: Định nghĩa Controller và các route	6
Bước 5: Cài đặt xử lý lỗi	10
GitHub:	13

Part 1

Bước 0: Cài đặt node và git

```
● PS D:\Doc\PTUDWeb> node -v  
v20.16.0  
● PS D:\Doc\PTUDWeb> git --version  
git version 2.46.0.windows.1
```

Bước 1: Tạo ứng dụng Node

Khởi tạo ứng dụng Node:

```

• PS D:\Doc\PTUDWeb> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (ptudweb)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author: BinhB2113327
license: (ISC)
About to write to D:\Doc\PTUDWeb\package.json:

{
  "name": "ptudweb",
  "version": "1.0.0",
  "main": "index.js",
  "directories": {
    "test": "test"
  },
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "BinhB2113327",
  "license": "ISC",
  "description": ""
}

Is this OK? (yes)

```

Cài đặt các package cần thiết:

```

PS D:\Doc\PTUDWeb> npm install express cors

added 67 packages, and audited 68 packages in 6s

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

found 0 vulnerabilities

```

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 (ví dụ như thư mục `node_modules`).

```
PS D:\Doc\PTUDWeb> npx gitignore node
Need to install the following packages:
gitignore@0.7.0
Ok to proceed? (y)

Created .gitignore file for flag type node.
```

Trong thư mục gốc dự án, chạy lệnh `git init`. Gõ tiếp `git status`, chúng ta sẽ thấy có 3 tập tin chưa được quản lý bởi git là `.gitignore`, `package-lock.json` và `package.json`.

```
PS D:\Doc\PTUDWeb> git init
Initialized empty Git repository in D:/Doc/PTUDWeb/.git/
PS D:\Doc\PTUDWeb> 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
test

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

Cho git quản lý 3 tập tin `.gitignore`, `package-lock.json` và `package.json` bằng lệnh sau đây:
`git add .gitignore, package-lock.json và package.json`

Gõ `git status` để kiểm tra lại:

```
PS D:\Doc\PTUDWeb> git add .gitignore package-lock.json package.json
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
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 D:\Doc\PTUDWeb> 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
```

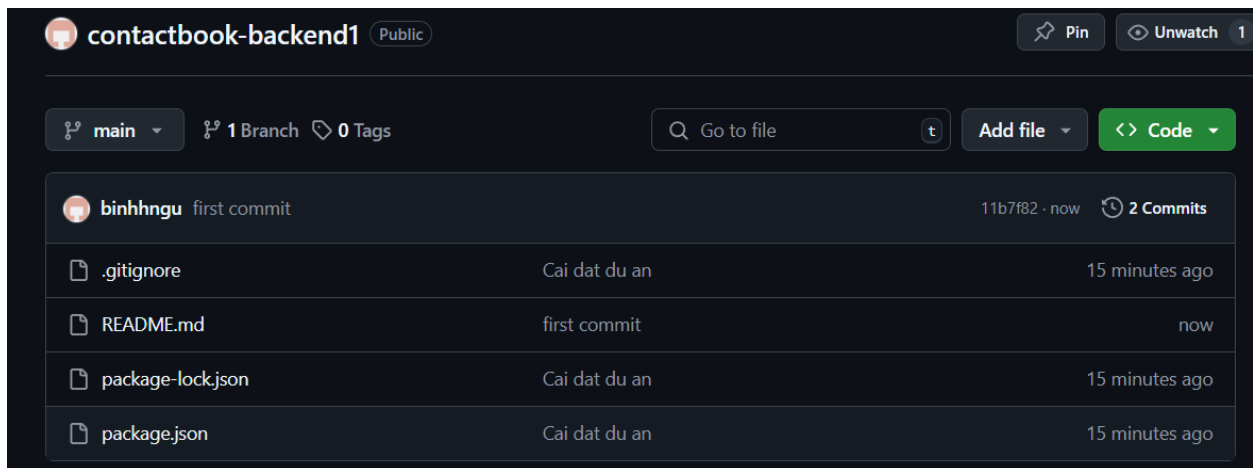
Thực thi `git commit -m "Cai dat du an"` để lưu các thay đổi. Tùy chọn `-m` trong câu lệnh cho phép tạo ghi chú cho commit này.

```

● PS D:\Doc\PTUDWeb> git commit -m "Cai dat du an"
[master (root-commit) b55b915] Cai dat du an
3 files changed, 944 insertions(+)
create mode 100644 .gitignore
create mode 100644 package-lock.json
create mode 100644 package.json

```

Đăng ký tài khoản và tạo dự án trên GitHub. Sau khi tạo, chúng ta upload dự án cục bộ lên GitHub như sau:



Bước 3: Cài đặt Express

Trong thư mục gốc của dự án, tạo tập tin app.js và server.js.

```

JS app.js  U X
JS app.js > ...
1  const express = require('express');
2  const cors = require('cors');
3
4  const app = express();
5
6  app.use(cors());
7  app.use(express.json());
8
9  app.get("/", (req, res) => {
10    res.json({message: "Welcome to contact book application."});
11  });
12
13  module.exports = app;

```

```
JS server.js U X
JS server.js > ...
1  const app = require("./app");
2  const config = require("./app/config");
3
4  const PORT = config.app.port;
5
6  app.listen(PORT, () => {
7    console.log(`Server is running on port ${PORT}.`);
8  });
```

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

```
app > config > JS index.js > ...
1  const config = {
2    app: {
3      port: process.env.PORT || 3000,
4    },
5  };
6
7  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.

```
Pretty-print ☒
{
  "message": "Welcome to contact book application."
}
```

Để dễ dàng hơn cho việc phát triển code, chúng ta có thể cài đặt thêm gói hỗ trợ nodemon. Gói nodemon này sẽ giám sát các thay đổi trên mã nguồn và tự động khởi động server:

```
● PS D:\Doc\PTUDWeb> npm install nodemon --save-dev
added 29 packages, and audited 97 packages in 4s

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

found 0 vulnerabilities
○ PS D:\Doc\PTUDWeb> 
```

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

```
"scripts": {
  "start": "nodemon server.js"
},
```

Lưu các thay đổi trên vào git:

```
● PS D:\Doc\PTUDWeb> git commit -m "Cai dat express chay thong bao xin chao"
[main e8f2e24] Cai dat express chay thong bao xin chao
```

Bước 4: Định nghĩa Controller và các route

Trong thư mục app/controllers, tạo tập tin contact.controller.js.


```
app > controllers > JS contact.controller.js > ...
1  exports.create = async (req, res) => {
2    res.send({message: 'create handler'})
3  };
4
5  exports.findAll = async (req, res) => {
6    res.send({message: 'findAll handler'})
7  };
8
9  exports.findOne = async (req, res) => {
10   res.send({message: 'findOne handler'})
11 };
12
13 exports.update = async (req, res) => {
14   res.send({message: 'update handler'})
15 };
16
17 exports.delete = async (req, res) => {
18   res.send({message: 'delete handler'})
19 };
20
21 exports.deleteAll = async (req, res) => {
22   res.send({message: 'deleteAll handler'})
23 };
24
25 exports.findAllFavorite = async (req, res) => {
26   res.send({message: 'findAllFavorite handler'})
27 };
```

Trong thư mục app/routes, tạo tập tin contact.route.js.

```

app > routes > JS contact.route.js > ...
1  const express = require("express");
2  const contacts = require("../controllers/contact.controller");
3
4  const router = express.Router();
5
6  router.route("/")
7    .get(contacts.findAll)
8    .post(contacts.create)
9    .delete(contacts.deleteAll);
10
11 router.route("/favorite")
12   .get(contacts.findAllFavorite);
13
14 router.route("/:id")
15   .get(contacts.findOne)
16   .put(contacts.update)
17   .delete(contacts.delete);
18
19 module.exports = router;

```

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

```

JS app.js > ...
1  const express = require("express");
2  const cors = require("cors");
3  const contactsRouter = require("./app/routes/contact.route");
4
5  const app = express();
6
7  app.use(cors());
8  app.use(express.json());
9
10 app.get("/", (req, res) => {
11   res.json({ message: "Welcome to contact book application." });
12 });
13
14 app.use("/api/contacts", contactsRouter);
15
16 module.exports = app;

```

Sử dụng ứng dụng Postman hoặc curl để kiểm tra các route vừa tạo.

The image displays three sequential screenshots of the Postman web interface, demonstrating how to configure and execute REST API requests.

First Screenshot (GET Request): The interface shows a GET request to the URL `http://localhost:3000/api/contacts`. The 'Query Params' section is empty. The 'Body' tab is selected, showing a JSON response in 'Pretty' view:

```
{  "message": "findAll handler"}
```

Second Screenshot (POST Request): The interface shows a POST request to the same URL. The 'Query Params' section is empty. The 'Body' tab is selected, showing a JSON response in 'Pretty' view:

```
{  "message": "create handler"}
```

Third Screenshot (GET Request): The interface shows a GET request to the URL `http://localhost:3000/api/contacts/favorite`. The 'Query Params' section is empty. The 'Body' tab is selected, showing a JSON response in 'Pretty' view:

```
{  "message": "findAllFavorite handler"}
```

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

```

• PS D:\Doc\PTUDWeb> git add -u
• PS D:\Doc\PTUDWeb> git add app/controllers app/routes
• PS D:\Doc\PTUDWeb> git commit -m "Định nghĩa các route cho tài nguyên Contact"
[main 6b9c7b0] Định nghĩa các route cho tài nguyên Contact
 3 files changed, 49 insertions(+)
 create mode 100644 app/controllers/contact.controller.js
 create mode 100644 app/routes/contact.route.js

```

Bước 5: Cài đặt xử lý lỗi

Tạo tập tin app/api-error.js.

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

```

const express = require('express');
const cors = require('cors');
const contactsRouter = require("./app/routes/contact.route");
const ApiError = require("./app/api-error");
const app = express();

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

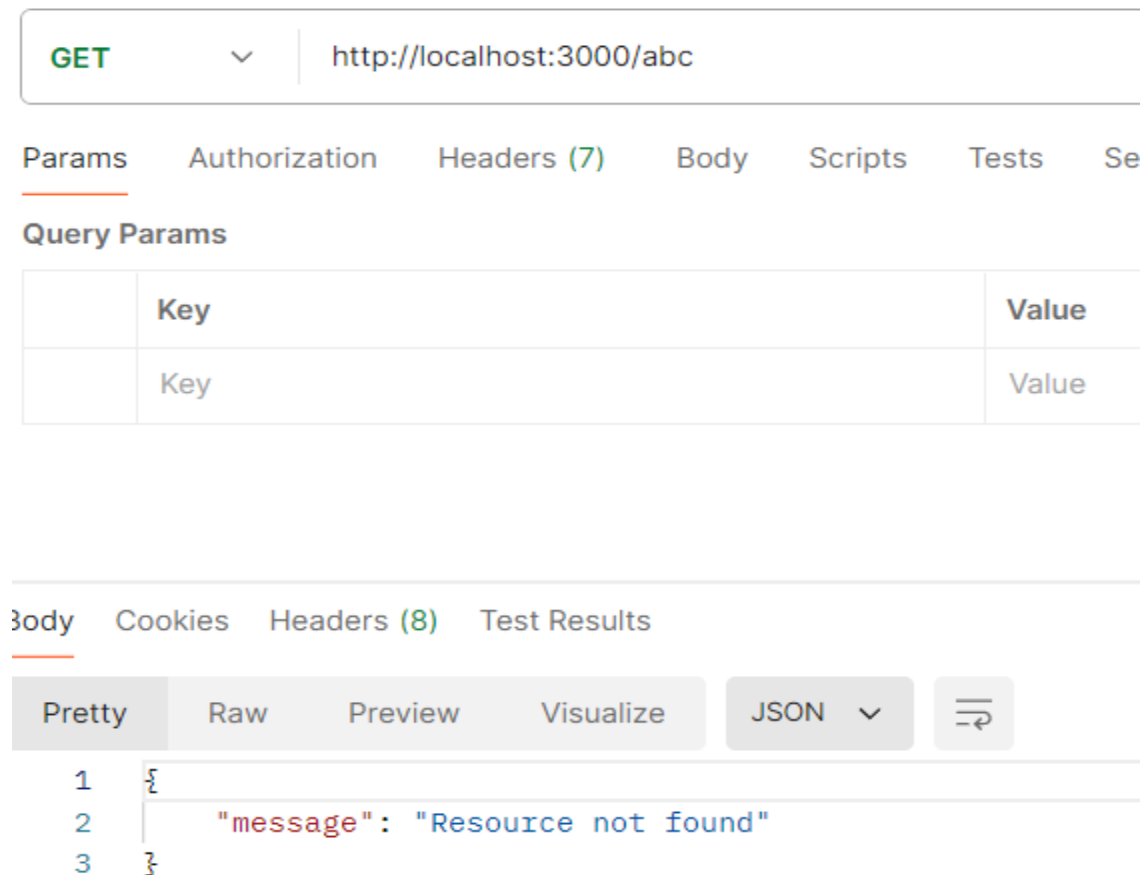
app.get("/", (req, res) => {
  res.json({ message: 'Welcome to contact book application.' });
});

// handle 404 response
app.use((req, res, next) => {
  // Code ở đây sẽ chạy khi không có route được định nghĩa nào
  // khớp với yêu cầu. Gọi next() để chuyển sang middleware xử lý lỗi
  return next(new ApiError(404, "Resource not found"));
});

// define error-handling middleware last, after other app.use() and routes calls
app.use((err, req, res, next) => {
  // Middleware xử lý lỗi tập trung.
  // Trong các đoạn code xử lý ở các route, gọi next(error)
  // sẽ chuyển về middleware xử lý lỗi này
  return res.status(err.statusCode || 500).json({
    message: err.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 D:\Doc\PTUDWeb> git add -u
• PS D:\Doc\PTUDWeb> git add app/api-error.js
• PS D:\Doc\PTUDWeb> git commit -m 'Cai dat xu ly loi'
[main 4f23940] Cai dat xu ly loi
2 files changed, 20 insertions(+), 2 deletions(-)
create mode 100644 app/api-error.js
```

Đẩy các thay đổi lên GitHub bằng lệnh: `git push origin master`.

```
PS D:\Doc\PTUDWeb> git push origin main
Enumerating objects: 25, done.
Counting objects: 100% (25/25), done.
Delta compression using up to 16 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (22/22), 6.24 KiB | 2.08 MiB/s, done.
Total 22 (delta 7), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (7/7), completed with 2 local objects.
To https://github.com/binhhngu/contactbook-backend1.git
11b7f82..4f23940  main -> main
```

 binhhngu Cai dat xu ly loi	4f23940 · 1 minute ago	 5 Commits
 app	Cai dat xu ly loi	1 minute ago
 .gitignore	Cai dat du an	5 hours ago
 README.md	first commit	5 hours ago
 app.js	Cai dat xu ly loi	1 minute ago
 package-lock.json	Cai dat express chay thong bao xin chao	23 minutes ago
 package.json	Cai dat express chay thong bao xin chao	23 minutes ago
 server.js	Cai dat express chay thong bao xin chao	23 minutes ago

GitHub:

<https://github.com/binhhngu/contactbook-backend1>