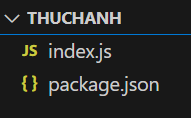
# Buổi 1

## Phần mềm

<https://drive.google.com/drive/folders/1zigkGPGyqzQqto4w9Io_TvDLmddtgcjp?usp=sharing>

## Làm quen môi trường Nodejs

1. Cài đặt Nodejs và Visual Code nếu **chưa có**
2. Tạo thư mục **THUCHANH**
3. Mở Visual Studio, chọn thư mục vừa chọn ở bước 1
   * File/Open folder/Chọn thư mục
4. Mở Terminal
   * View/Terminal
5. Trong Terminal Visual Code, gõ lệnh (tạo file package.json):
   * npm init
   * gõ enter cho các yêu cầu tiếp tiếp
   * bước cuối: Is this OK? (yes), **nhập y**
   * **Kết quả: => Có file package.json**
6. Tạo file index.js, kết quả như hình



* + Nhập nội dung sau vào **index.js**

var http = require('http');

http.createServer(function (req, res) {

res.writeHead(200, {'Content-Type': 'text/html'});

res.end('Hello KTPM0121');

}).listen(8080);

1. Thực thi file index.js
   * Tại Terminal, gõ lệnh: **node index.js**
   * Mở trình duyệt web, nhập URL: [**http://localhost:8080/**](http://localhost:8080/)
2. Sửa lại 2 dòng trên file index.js

res.writeHead(200, {'Content-Type': 'text/html***;charset=utf-8***'});

res.end('Hello KTPM0121, ***chúc bạn thành công với Nodejs***');

1. Thực thi lại file index.js
   * Trong Terminal, nhấn tổ hợp **phím C** để thoát
   * Gõ lại lệnh: **node index.js**
   * **Refresh** lại trang web trên trình duyệt web, nhận kết quả mới

## Xây dựng module date.js

1. Tạo file date.js, nhập nội dung sau:

const myDateTime = (str="Ngày hiện tại: ") => {

return str + Date()

}

module.exports=myDateTime

1. Sử dụng module date.js ở file index.js

var http = require('http');

**var date=require("./date")**

http.createServer(function (req, res) {

res.writeHead(200, {'Content-Type': 'text/html; charset=utf-8'});

**res.write(date()+ “<br>”);**

res.write('Hello KTPM0121, chúc bạn thành công với Nodejs');

res.end();

}).listen(8080);

1. Thực thi lại file index.js
2. Mở trình duyệt web, nhập URL: [**http://localhost:8080/**](http://localhost:8080/)

## Xây dựng module getURL.js

1. Tạo file getURL.js, nhập nội dung sau:

var url=require('url')

const getPath=(req)=>{

return req.url;

}

const getParamsURL=(req)=>{

let urlData = url.parse(req.url, true);

return JSON.stringify(urlData.query);

}

module.exports={getPath,getParamsURL}

1. Sử dụng module getURL.js ở file index.js

var http = require('http');

var date=require("./date")

**var getURL = require("./getURL")**

http.createServer(function (req, res) {

res.writeHead(200, {'Content-Type': 'text/html;charset=utf-8'});

res.write(date()+"<br>");

**res.write(getURL.getPath(req)+"<br>");**

**res.write(getURL.getParamsURL(req)+"<br>");**

res.write('Hello KTPM0121, chúc bạn thành công với Nodejs');

res.end();

}).listen(8080);

1. Thực thi lại file index.js
2. Mở trình duyệt web, nhập URL: **http://localhost:8080/test.htm?animal=cat&color=black**

## Viết theo chuẩn ES6

1. Chỉnh sửa tệp index.js, date.js, getURL.js về chuẩn ES6
   * Thay thế lệnh **require** thành lệnh **import**
   * Sửa **module.exports** thành **export defaut**
2. Thực thi lại file index.js
3. Cài đặt babel (chạy được ES6)
   * npm install --save-dev @babel/core @babel/cli @babel/preset-env @babel/node
     + Xem kết quả cài đặt trong file package-lock.json
   * Cấu hình babel
     + Tạo babel.config.json với nội dung

{

"presets": [

[

"@babel/preset-env",

{

"targets": {

"edge": "17",

"firefox": "60",

"chrome": "67",

"safari": "11.1"

},

"useBuiltIns": "usage",

"corejs": "3.6.5"

}

]

]

}

* + - Mở file package.json, hiệu chỉnh như dòng code in đậm:

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1",

**"start":"babel-node index.js"**

},

1. Thực thi lại file index.js với lệnh: **npm start**

## Cấu hình thêm các thư viện cần thiết cho môi trường Nodejs

1. Cài đặt dotenv
   * npm I dotenv
   * Tạo file .env với nội dung

#cong chay

PORT=3000

1. Cài đặt nodemon
   * Cài đặt bằng lệnh: npm i nodemon
   * Chỉnh sửa file package.json

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1",

**"start":"nodemon --exec babel-node index.js"**

},

## Xây dựng ứng dụng với framework express

1. Cài đặt Express
   * npm i express
2. Tạo file **server.js** với nội dung

import express from 'express'

import dotenv from 'dotenv/config'

const app = express()

const port=process.env.PORT

app.get('/', **(req, res) => {**

**res.send('Hello World!')**

**}**)

app.listen(port, () => {

console.log(`Example app listening on port ${port}`)

})

1. Chỉnh sửa file package.json

**"start":"nodemon --exec babel-node server.js"**

1. Thực thi file server.js, xem kết quả
2. Mở trình duyệt web, nhập URL: **localhost:3000**
3. Thêm vào server.js nội dung sau

app.get('/about', (req, res) => {

res.send('Hello World!. Page about')

})

1. Mở trình duyệt web, nhập URL: **localhost:3000/about**
2. Gọi module date.js trong file server.js, hiển thị thời gian khi người dùng nhập URL: **http://localhost:3000/date**
3. Gọi module getURL.js trong file server.js, hiển thị đường dẫn khi người dùng nhập URL: **http://localhost:3000/geturl?cat&black**

## View Engine

1. Cài đặt view engine EJS
   * npm i ejs
   * Tạo file viewEngine.js để chứa cấu hình ejs

const viewEngine=(app)=>{

app.set("view engine","ejs")

app.set("views","./")

}

export default viewEngine

1. Sử dụng viewEngine.js trong server.js
2. Tạo file test.ejs, chứa nội dung

Test view engine ejs

1. Xây dựng route trong file server.js để thực thi file test.ejs

app.get('/ejs', (req, res) => {

res.render("test")

})

1. Mở trình duyệt và nhập URL: **http://localhost:3000/ejs**
2. Xây dựng thêm 1 số file ejs: home.ejs, about.ejs tương với các route đã dựng trước
   * Route 1
     + path: /
     + render: home
   * Route 2:
     + path: /about
     + render: about

# **BUỔI 2**

Lấy source buổi 1 của Nodejs đã đưa lên git hub để làm tiếp buổi thực hàng số 2

## Dựng layout cho trang chính: thiết kế trang home.ejs gồm các trang

* + Thiết kế header.ejs chứa menu hệ thống
    - Dùng cấu trúc Navbar của Bootrap 5 thiết kế, mỗi menu chứa liên kết tương ứng với các route đã thiết kế
  + Thiết kế trang footer.ejs chứa thông tin của người thiết kế website
  + Thiết kế trang main.ejs chứa thông tin mặc định của website tương ứng với path: /

|  |
| --- |
| **header.ejs** |
| **KHUNG CHÍNH**  **{ nội dung tùy biến, phụ thuộc vào route được truy cập }** |
| **footer.ejs** |

## Xây dựng file webRoute để điều hướng cho web

## Xây dựng các Controller ứng với các route path như sau:

* + Route path: /
    - Controller: HomeController
    - Khung chính render: main.ejs
  + Route path: /about
    - Controller: AboutController
    - Khung chính render: about.ejs
  + Route: /contact
    - Controller: ContactController
    - Khung chính render: contact.ejs

## Thiết kế cơ sở dữ liệu có bảng như sau:

* + - users(**username,** password, fullname, address, sex, email)

## Xây dựng hoàn chỉnh các chức năng cơ bản trên bảng users theo mô hình MVC

* + Hiển thị danh sách người dùng
    - Xem, sửa, xóa người dùng cụ thể như mẫu



* + Thêm người dùng mới
  + Hoàn chỉnh file header.ejs để điều hướng cho các route đã thiết kế