

I. Buổi 1

I.1. Phần mềm

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

I.2. Làm quen môi trường Nodejs

Bước 0. Cài đặt Nodejs và Visual Code nếu **chưa có**

Bước 1. Tạo thư mục **THUCHANH**

Bước 2. Mở Visual Studio, chọn thư mục vừa chọn ở bước 1

- File/Open folder/Chọn thư mục

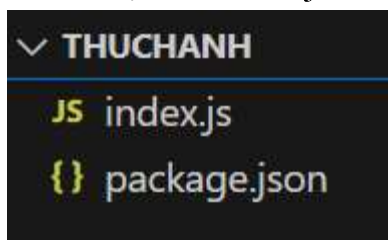
Bước 3. Mở Terminal

- View/Terminal

Bước 4. 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**

Bước 5. 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);
```

Bước 6. Thực thi file index.js

- Tại Terminal, gõ lệnh: **node index.js**
- Mở trình duyệt web, nhập URL: <http://localhost:8080/>

Bước 7. 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');
```

Bước 8. 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

I.3. Xây dựng module date.js

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

```
const myDateTime = (str="Ngày hiện tại: ") => {  
  return str + Date()  
}  
module.exports=myDateTime
```

Bước 2. 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);
```

Bước 3. Thực thi lại file index.js

Bước 4. Mở trình duyệt web, nhập URL: <http://localhost:8080/>

I.4. Xây dựng module getURL.js

Bước 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}

```

Bước 2. 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);

```

Bước 3. Thực thi lại file index.js

Bước 4. Mở trình duyệt web, nhập URL:

http://localhost:8080/test.htm?animal=cat&color=black

I.5. Viết theo chuẩn ES6

Bước 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 default**

Bước 2. Thực thi lại file index.js

Bước 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"
},

```

Bước 4. Thực thi lại file index.js với lệnh: **npm start**

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

Bước 1. Cài đặt dotenv

- ✓ npm i dotenv
- ✓ Tạo file .env với nội dung


```

#cong chay
PORT=3000

```

Bước 2. 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"
},

```

I.7. Xây dựng ứng dụng với framework express

Bước 1. Cài đặt Express

✓ `npm i express`

Bước 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}`)
})
```

Bước 3. Chỉnh sửa file package.json

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

Bước 4. Thực thi file server.js, xem kết quả

Bước 5. Mở trình duyệt web, nhập URL: **localhost:3000**

Bước 6. Thêm vào server.js nội dung sau

```
app.get('/about', (req, res) => {
  res.send('Hello World!. Page about')
})
```

Bước 7. Mở trình duyệt web, nhập URL: **localhost:3000/about**

Bước 8. Gọi module date.js trong file server.js, hiển thị thời gian khi người dùng nhập URL: **http://localhost:3000/date**

Bước 9. Gọi module getURL.js trong file server.js, hiển thị đường dẫn khi người dùng nhập URL: **http://localhost:3000/geturl?cat&black**

I.8. View Engine

Bước 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
```

Bước 2. Sử dụng viewEngine.js trong server.js

Bước 3. Tạo file test.ejs, chứa nội dung

```
Test view engine ejs
```

Bước 4. Xây dựng route trong file server.js để thực thi file test.ejs

```
app.get('/ejs', (req, res) => {  
  res.render("test")  
})
```

Bước 5. Mở trình duyệt và nhập URL: **http://localhost:3000/ejs**

Bước 6. 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