



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

Lập trình javaScript

Xây dựng Dịch vụ với NodeJS

<http://csc.edu.vn/laptrinh/>

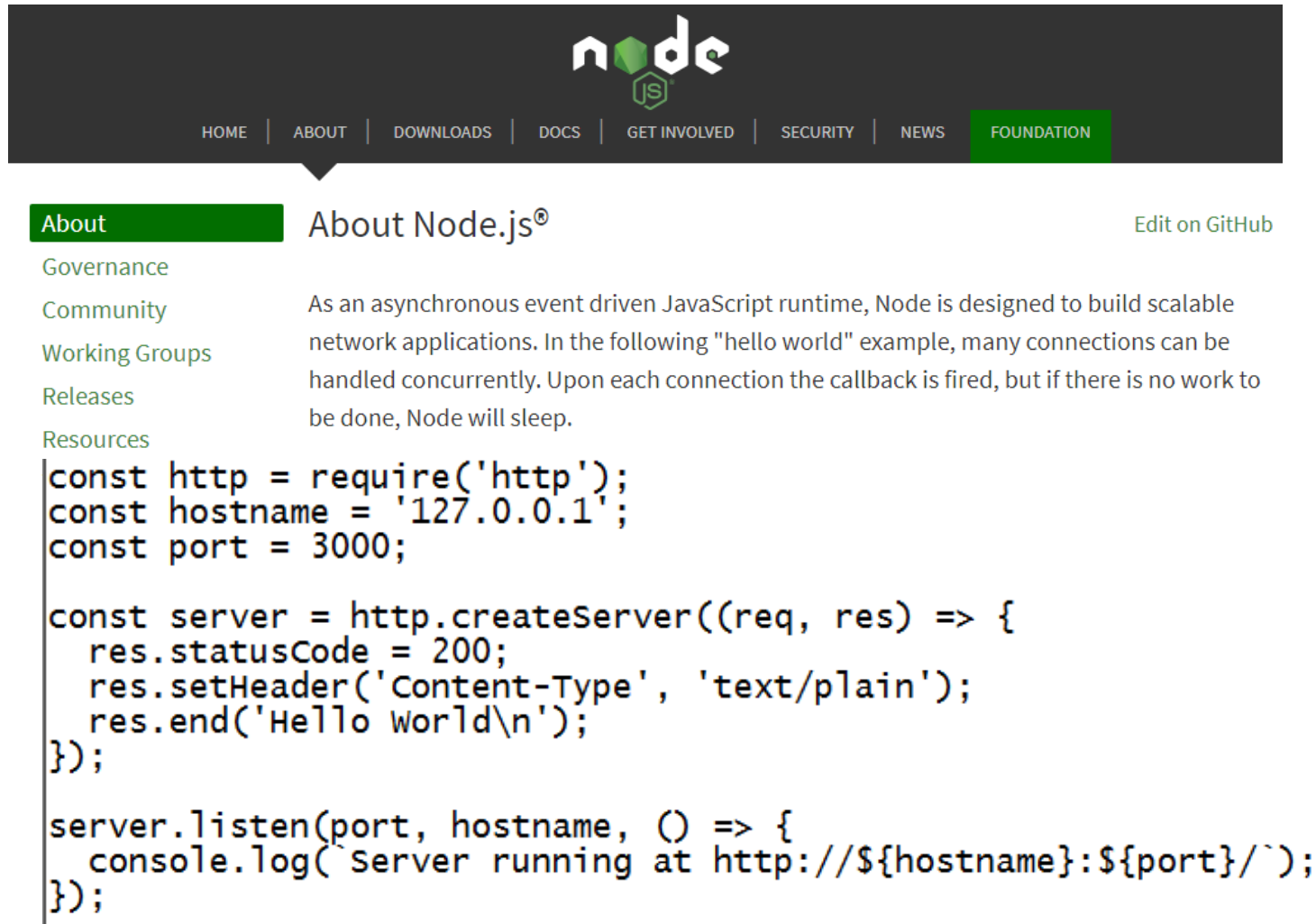
- ☐ Giới thiệu
- ☐ Xây dựng dịch vụ
- ☐ Minh họa

Giới thiệu

- Là thư viện JavaScript được dùng để xây dựng ứng dụng web xử lý phía Server hay các Dịch vụ dữ liệu.
- Thực thi trên nhiều hệ điều hành khác nhau: Window, Linux, OSX
- Hỗ trợ tốt với các ứng dụng thời gian thực (realtime)



Tài liệu: <https://nodejs.org>



The screenshot shows the Node.js website. At the top is a dark navigation bar with the Node.js logo and links: HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, NEWS, and FOUNDATION. The 'ABOUT' link is highlighted in green. Below the navigation bar, the 'About' section is active, showing the title 'About Node.js®' and a link to 'Edit on GitHub'. A list of links on the left includes Governance, Community, Working Groups, Releases, and Resources. The main content area describes Node.js as an asynchronous event-driven JavaScript runtime and includes a code block for a 'hello world' example.

About Node.js® [Edit on GitHub](#)

Governance

Community

Working Groups

Releases

Resources

As an asynchronous event driven JavaScript runtime, Node is designed to build scalable network applications. In the following "hello world" example, many connections can be handled concurrently. Upon each connection the callback is fired, but if there is no work to be done, Node will sleep.

```
const http = require('http');
const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

❑ Ưu điểm

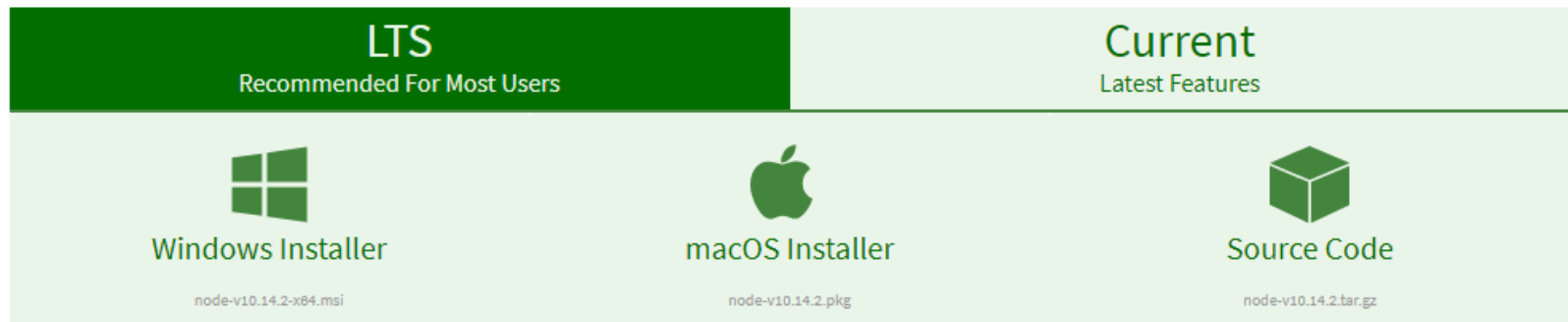
- Tốc độ xử lý nhanh
- Khả năng mở rộng cao
- Hỗ trợ lập trình bất đồng bộ

❑ Khuyết điểm

- Vì NodeJS thường được dùng chung với CSDL NoSQL (VD: MongoDB) nên nếu với lập trình viên đã quen với các kiểu lập trình trước sẽ tương đối khó khăn
- Khi xử lý dữ liệu lớn, NodeJS sẽ tốn rất nhiều tài nguyên

❑ Cài đặt

- Tải NodeJS theo link: <https://nodejs.org/en/download/>



Windows Installer (.msi)

Windows Binary (.zip)

macOS Installer (.pkg)

macOS Binary (.tar.gz)

Linux Binaries (x64)

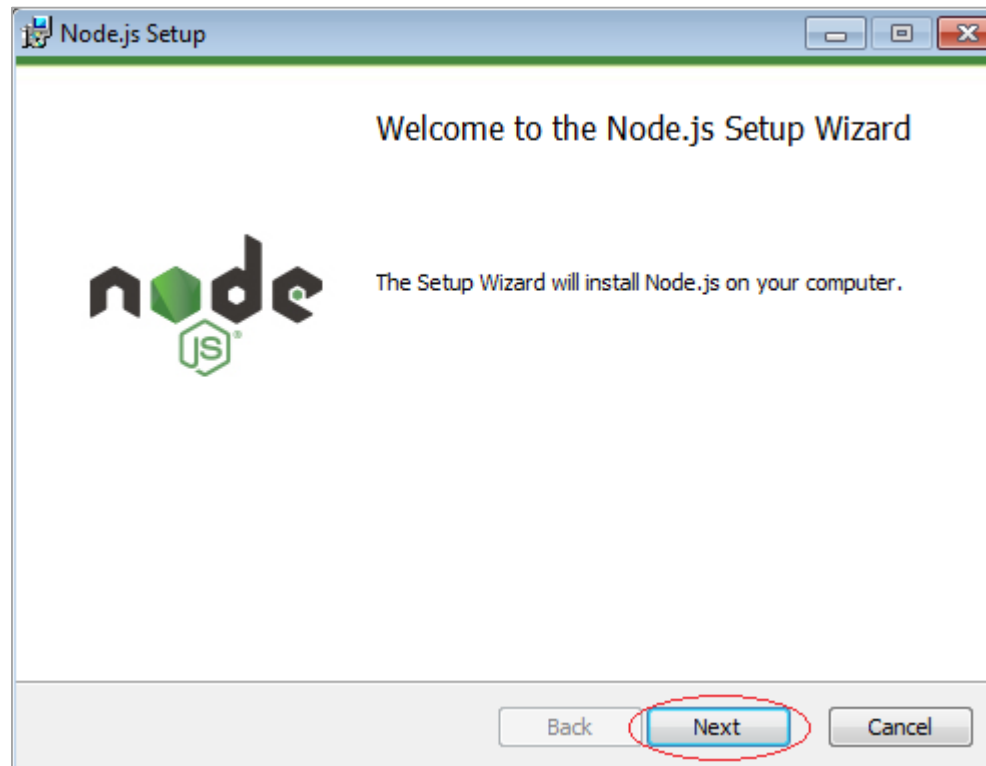
Linux Binaries (ARM)

Source Code

32-bit		64-bit	
32-bit		64-bit	
64-bit			
64-bit			
64-bit			
ARMv6	ARMv7		ARMv8
node-v10.14.2.tar.gz			

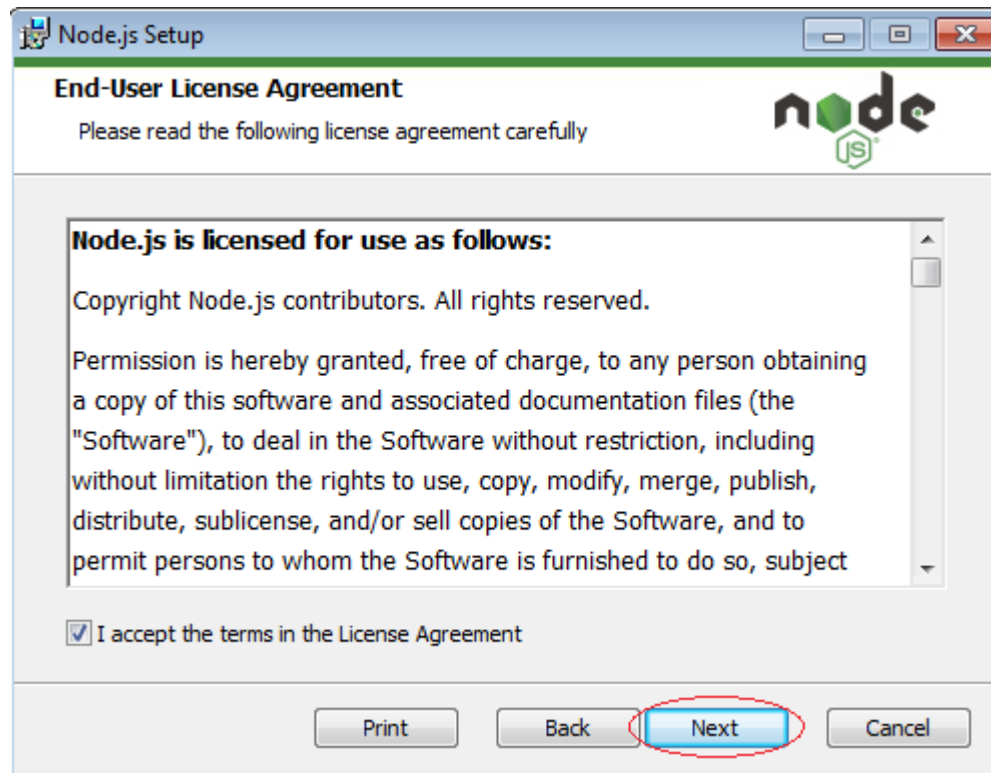
❑ Cài đặt

- Thực thi tập tin cài đặt



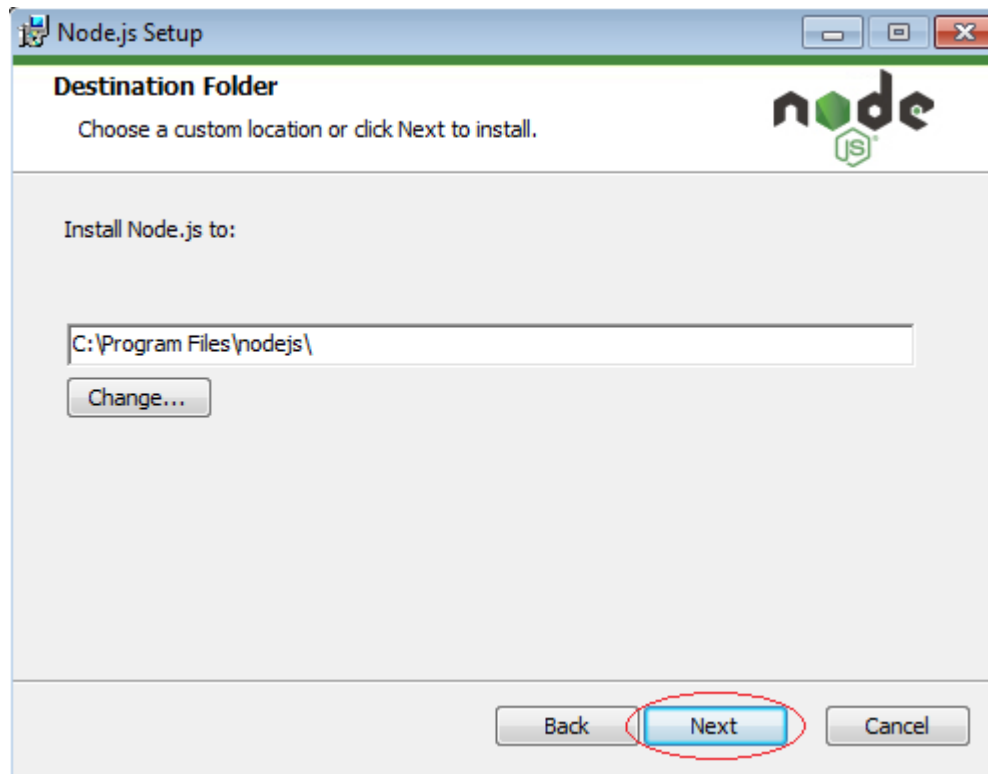
❑ Cài đặt

- Xác thực điều khoản người dùng



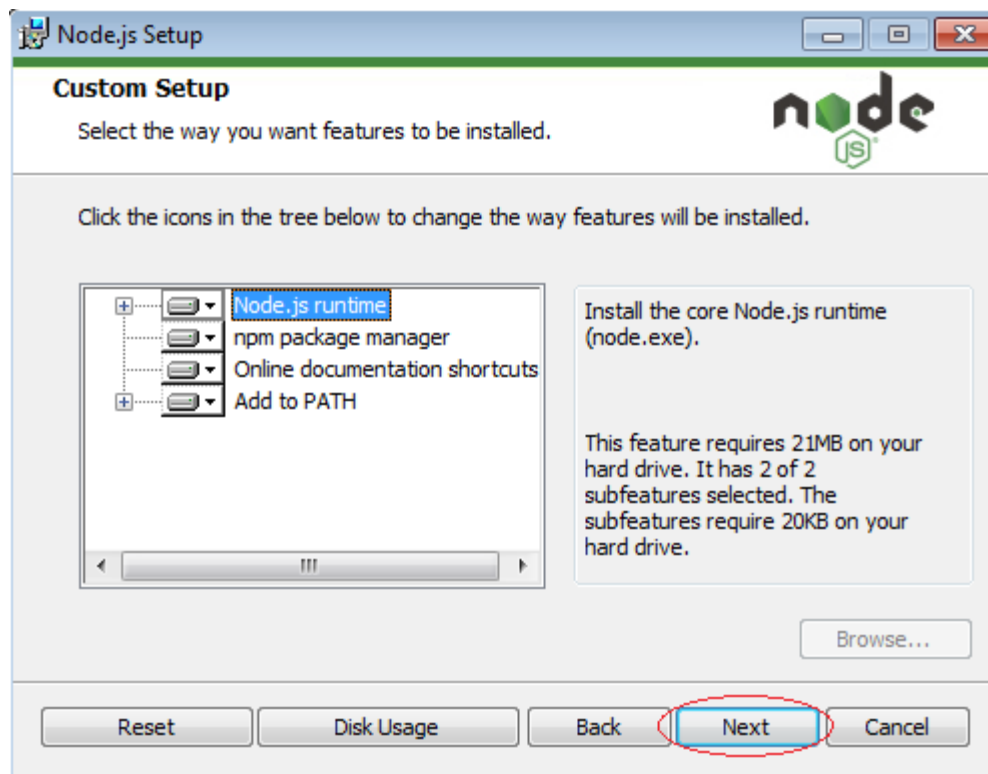
❑ Cài đặt

- Chọn thư mục cài đặt



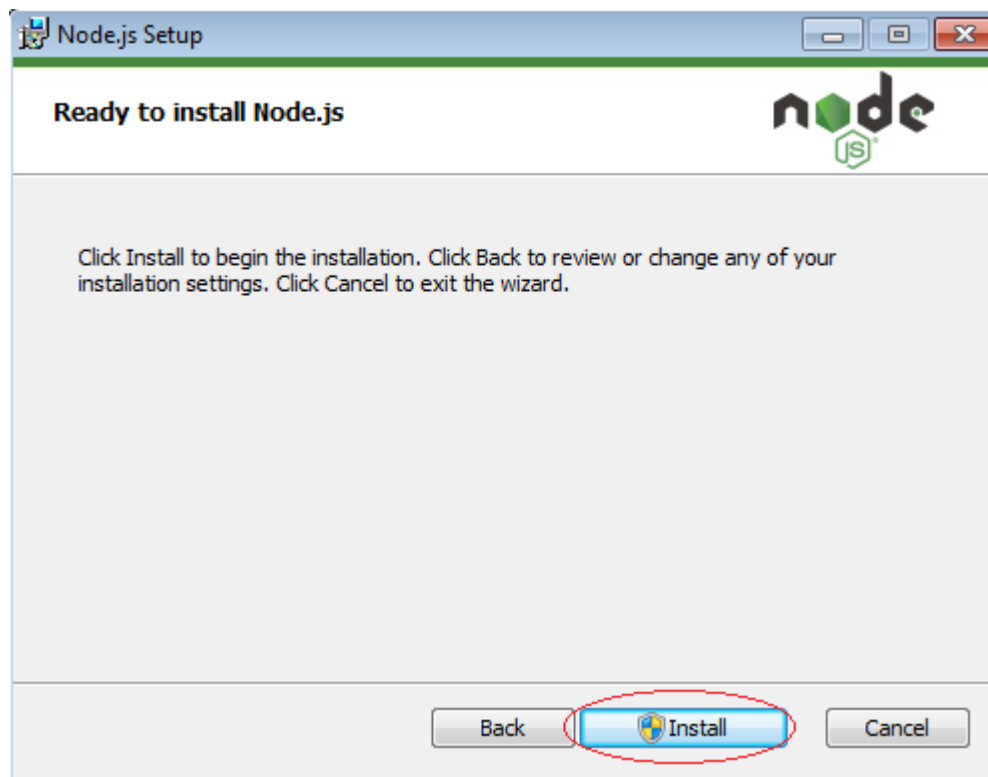
❑ Cài đặt

- Chọn thư viện cần cài đặt



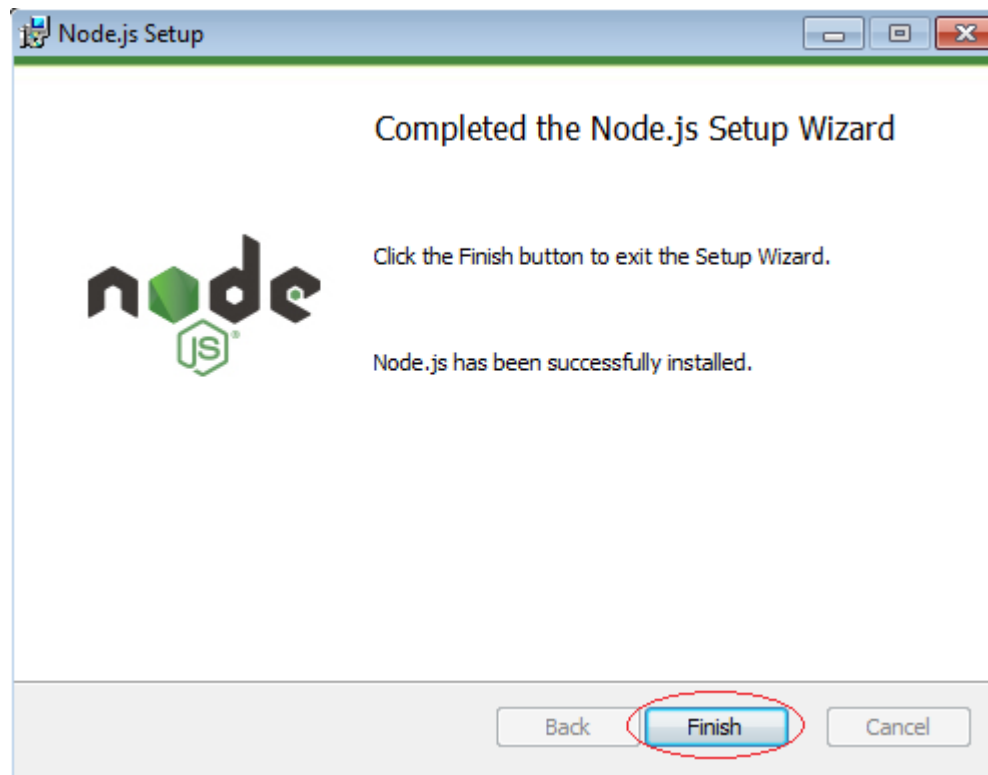
❑ Cài đặt

- Tiến hành cài đặt



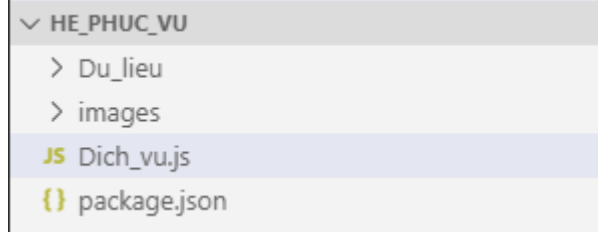
❑ Cài đặt

- Hoàn tất cài đặt



❑ Tạo ứng dụng

- Tạo thư mục ứng dụng



- Tạo tập tin package.json

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

D:\He_phuc_vu>npm init -y
```

- Xây dựng Dịch vụ

- ☐ Cấu hình dịch vụ
- ☐ Hệ phục vụ
- ☐ Hệ khách

Cấu hình dịch vụ



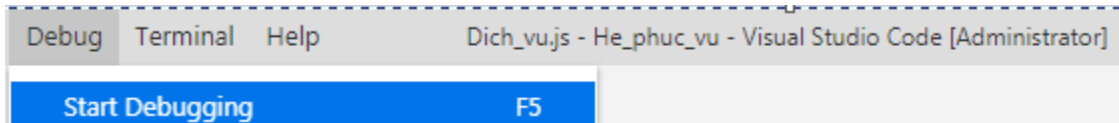
- Phương thức `createServer`: định nghĩa dịch vụ
- Thuộc tính `listen`: khai báo port khi sử dụng
- Ví dụ: xây dựng dịch vụ thực thi trên port 8080

```
JS Dich_vu.js > ...
1  var http = require("http");
2  var port = 8080;
3  var Dich_vu = http.createServer(
4    (Yeu_cau, Dap_ung) => {
5      let Ket_qua = "Chào bạn đến NodeJS";
6      Dap_ung.writeHead(200, { "Content-Type": "text/json; charset=utf-8" });
7      Dap_ung.end(Ket_qua);
8    }
9  )
10 Dich_vu.listen(port);
```

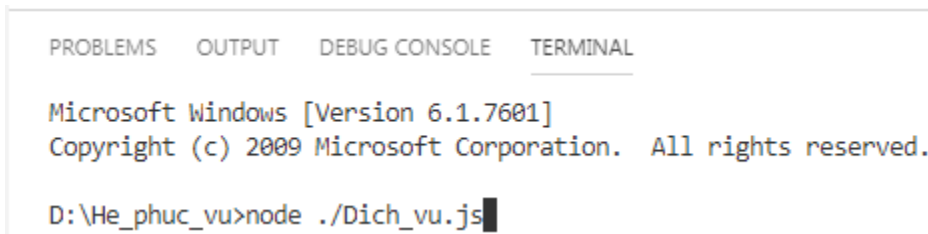
Cấu hình dịch vụ



- Thực thi Dịch vụ
 - Debug



- Node



Cấu hình dịch vụ



- Thực thi Dịch vụ

- `npm start`

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
```

```
D:\He_phuc_vu>npm start
```

```
> he_phuc_vu@1.0.0 start D:\He_phuc_vu
> node ./Dich_vu.js
```

- Trong tập tin: `package.json`

```
{ } package.json > ...
```

```
1  {
2    "name": "he_phuc_vu",
3    "version": "1.0.0",
4    "description": "Xay dung Dich vu",
5    "main": "index.js",
6    "scripts": {
7      "start": "node ./Dich_vu.js"
8    },
9    "author": "T3h",
10   "license": "ISC"
11 }
```



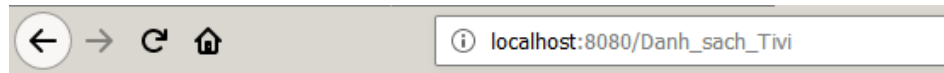
❑ Phương thức đọc dữ liệu từ tập tin JSON

- Ví dụ: xây dựng phương thức đọc danh sách Tivi

```
JS Dich_vu.js > ...
1  var http = require("http");
2  var fs = require("fs");
3  var port = 8080;
4
5  var Dich_vu = http.createServer(
6    (Yeu_cau, Dap_ung) => {
7      if (Yeu_cau.method == "GET") {
8        if (Yeu_cau.url == "/Danh_sach_Tivi") {
9          let Ket_qua = "";
10         Ket_qua = fs.readFileSync('./Du_lieu/Danh_sach_Tivi.json', 'utf8');
11         Dap_ung.writeHead(200, { "Content-Type": "text/json; charset=utf-8" });
12         Dap_ung.end(Ket_qua);
13       }
14     }
15   )
16 )
17 Dich_vu.listen(port);
```

❑ Phương thức đọc dữ liệu từ tập tin JSON

- Kết quả: phương thức đọc danh sách Tivi



```
[
  {
    "Ten": "Smart Tivi Cong Samsung 49 inch",
    "Ma_so": "TIVI 1",
    "Hinh": "http://10.0.3.2:8081/images/TIVI_1.png",
    "Don_gia": 22900000,
    "Danh_sach_Ban_hang": [
      {
        "Ma_so": "PB_1",
        "Ngay": "2018-10-10",
        "So_luong": 2,
        "Don_gia": 22900000
      },
      {
        "Ma_so": "PB_2",
        "Ngay": "2018-10-10",
        "So_luong": 1,
        "Don_gia": 22900000
      }
    ]
  },
  {
    "Ten": "Internet Tivi Sony 4K 43 inch",
    "Ma_so": "TIVI 4",
    "Hinh": "http://10.0.3.2:8081/images/TIVI_4.png",
    "Don_gia": 14500000,
    "Danh_sach_Ban_hang": []
  },
  {
    "Ten": "Internet Tivi Samsung 32 inch",
    "Ma_so": "TIVI 8",
    "Hinh": "http://10.0.3.2:8081/images/TIVI_8.png",
    "Don_gia": 6890000,
    "Danh_sach_Ban_hang": []
  },
]
```

❑ Phương thức ghi dữ liệu vào tập tin JSON

- Ví dụ: xây dựng phương thức cập nhật đơn giá Tivi, với thông tin nhận vào là đối tượng Tivi cần cập nhật

```
Dich_vu.js > ...
1  var http= require("http");
2  var fs= require("fs");
3  var url = require('url');
4  var port=8080;
5  var Dich_vu = http.createServer(
6    (Yeu_cau, Dap_ung)=>{
7      if(Yeu_cau.method=="GET"){
8        if(Yeu_cau.url== "/Danh_sach_Tivi"){
9          let Ket_qua="";
10         Ket_qua = fs.readFileSync('./Du_lieu/Danh_sach_Tivi.json', 'utf8');
11         Dap_ung.writeHead(200, {"Content-Type": "text/json; charset=utf-8"});
12         Dap_ung.end(Ket_qua);
13       }
14     }
15     else if(Yeu_cau.method == "POST"){
16       var Noi_dung_Nhan = '';
17       Yeu_cau.on('data', function (data) {
18         Noi_dung_Nhan += data;
19       });
20       if(Yeu_cau.url=="/Cap_nhat_Tivi"){
21         Yeu_cau.on('end', function(){
22           let Doi_tuong =JSON.parse(Noi_dung_Nhan);
23           let Ket_qua={"Noi_dung":"Lỗi cập nhật"};
24           let Chuoi_Danh_sach_Tivi=fs.readFileSync("./Du_lieu/Danh_sach_Tivi.json","utf8");
25           let Danh_sach_Tivi=JSON.parse Chuoi_Danh_sach_Tivi);
26           let Tivi = Danh_sach_Tivi.find(item =>item.Ma_so==Doi_tuong.Ma_so);
27           Tivi.Don_gia=Doi_tuong.Don_gia;
28           fs.writeFileSync("./Du_lieu/Danh_sach_Tivi.json",JSON.stringify(Danh_sach_Tivi),"utf8");
29           Ket_qua.No_i_dung=true;
30           Dap_ung.end(JSON.stringify(Ket_qua));
31         });
32       }
33     }
34   }
35 )
36 Dich_vu.listen(port);
```

❑ Hiện thị image

- Ví dụ: xây dựng phương thức hiện thị hình cho dịch vụ

```
var Dich_vu = http.createServer(  
  (Yeu_cau, Dap_ung) => {  
    if (Yeu_cau.method == "GET") {  
      if (Yeu_cau.url == "/Danh_sach_Tivi") {  
        let Ket_qua = "";  
        Ket_qua = fs.readFileSync('./Du_lieu/Danh_sach_Tivi.json', 'utf8');  
        Dap_ung.writeHead(200, { "Content-Type": "text/json; charset=utf-8" });  
        Dap_ung.end(Ket_qua);  
      } else if (Yeu_cau.url.match("\.png$")) {  
        var imagePath = `images/${Yeu_cau.url}`;  
        if (fs.existsSync(imagePath)) {  
          let fileStream = fs.createReadStream(imagePath);  
          Dap_ung.writeHead(200, { "Content-Type": "image/png" });  
          fileStream.pipe(Dap_ung);  
        } else {  
          imagePath = `images/noImage.png`;  
          let fileStream = fs.createReadStream(imagePath);  
          Dap_ung.writeHead(200, { "Content-Type": "image/png" });  
          fileStream.pipe(Dap_ung);  
        }  
      }  
    }  
  })  
);
```

❑ Hiển thị image

- Ví dụ: Gọi phương thức hiển thị hình cho dịch vụ



- Giáo viên minh họa