

React Advance

Bài 1: Giới thiệu NodeJS



Nội dung

1. Giới thiệu NodeJS
2. Cài đặt môi trường
3. Function Expression
4. Modules Patterns
5. Xử lý File với NodeJS
6. Xử lý Folder với NodeJS
7. Dựng Web Server với NodeJS



1. Giới thiệu NodeJS



Giới thiệu NodeJS

- NodeJS là một nền tảng được xây dựng trên V8 Engine của Google.
- Node hoạt động dựa trên nguyên lý hướng sự kiện (event-driven) bất đồng bộ (*asynchronous*) của JavaScript runtime.
- NodeJS được thiết kế để xây dựng các ứng dụng mạng có thể mở rộng.
- Trang chủ NodeJS <https://nodejs.org/>



Giới thiệu NodeJS

```
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');
});

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



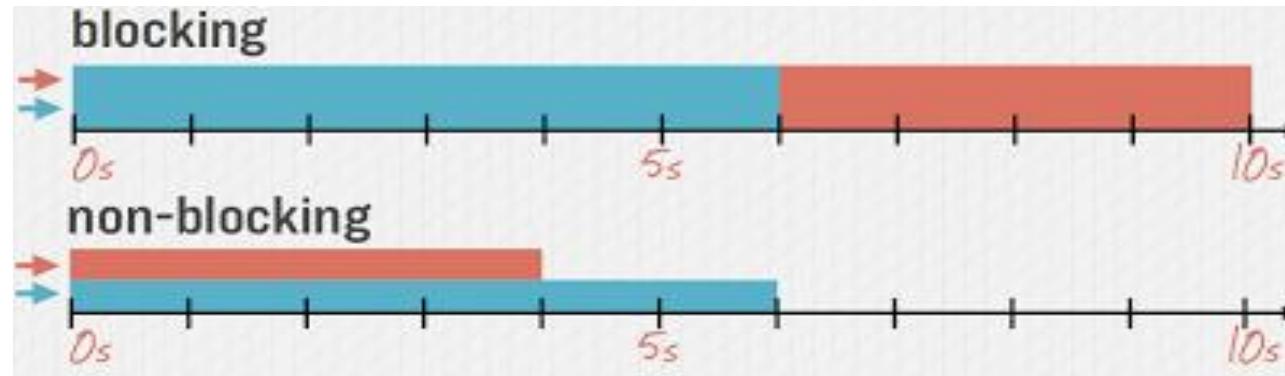
Blocking I/O vs Non-Blocking I/O

Blocking I/O

```
var contents = fs.readFileSync('hello.txt');
console.log(contents);
console.log('Thực hiện công việc khác');
```

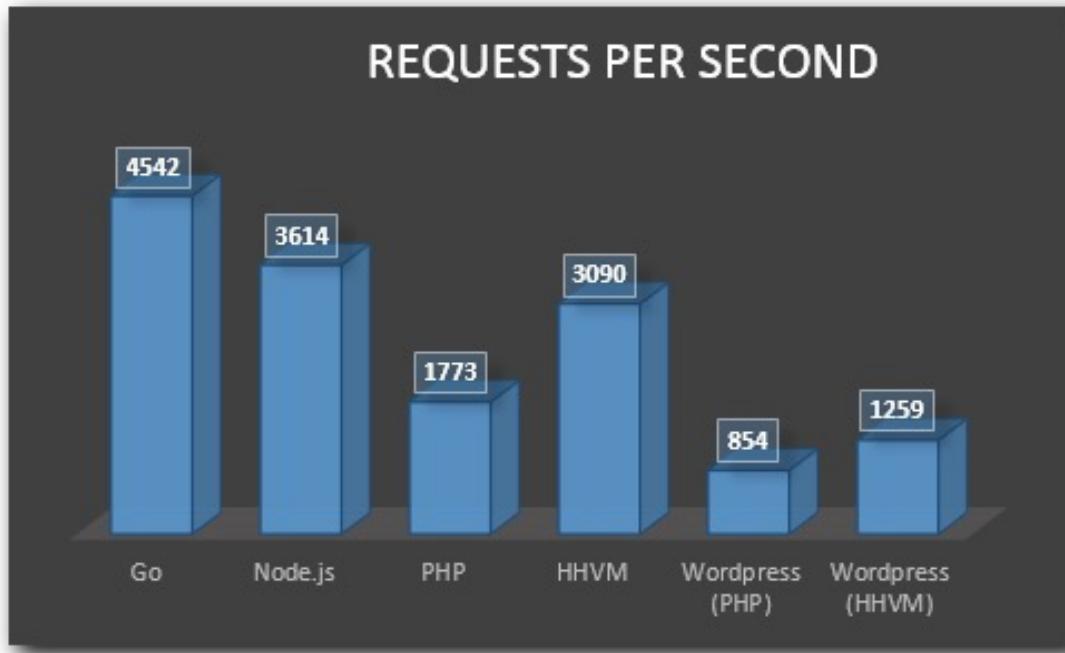
Non-Blocking I/O

```
fs.readFile('hello.txt', function(contents){
  console.log(contents);
});
console.log('Thực hiện công việc khác');
```

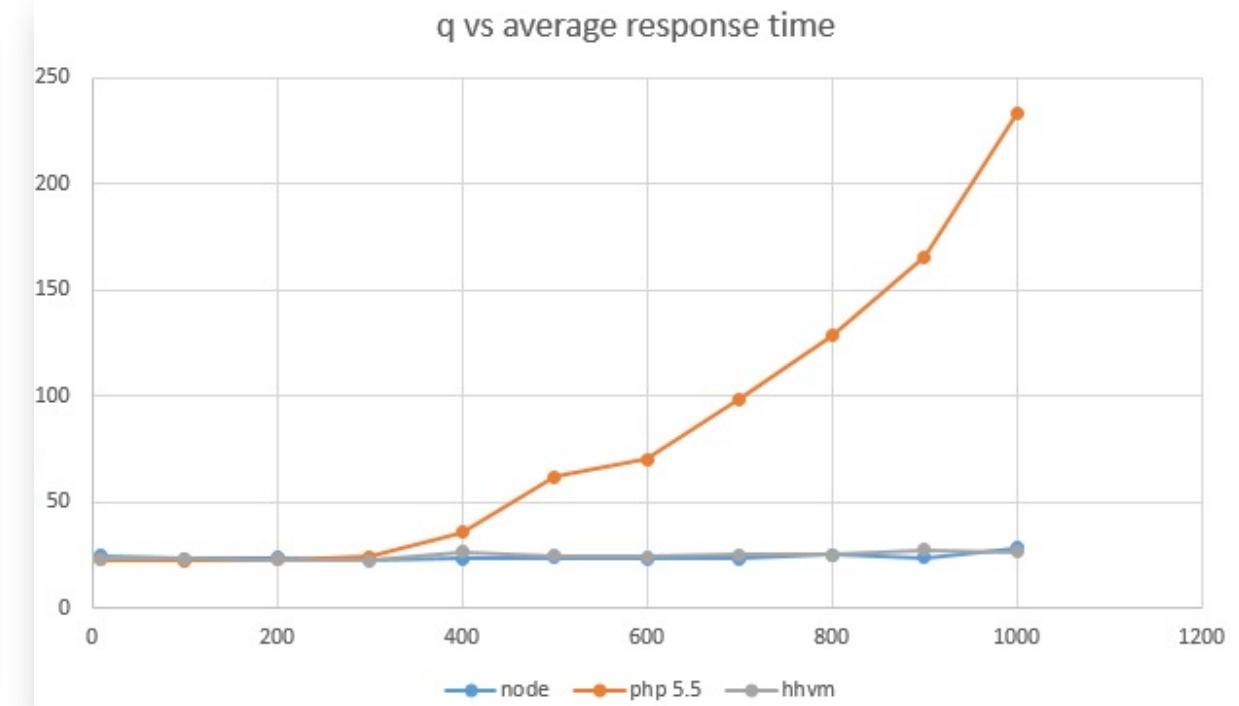




Hiệu năng của NodeJS



Simple HTTP Requests



HTTP + CPU tasks

Ref: <http://www.hostingadvice.com/blog/comparing-node-js-vs-php-performance/>



Hiệu năng của NodeJS

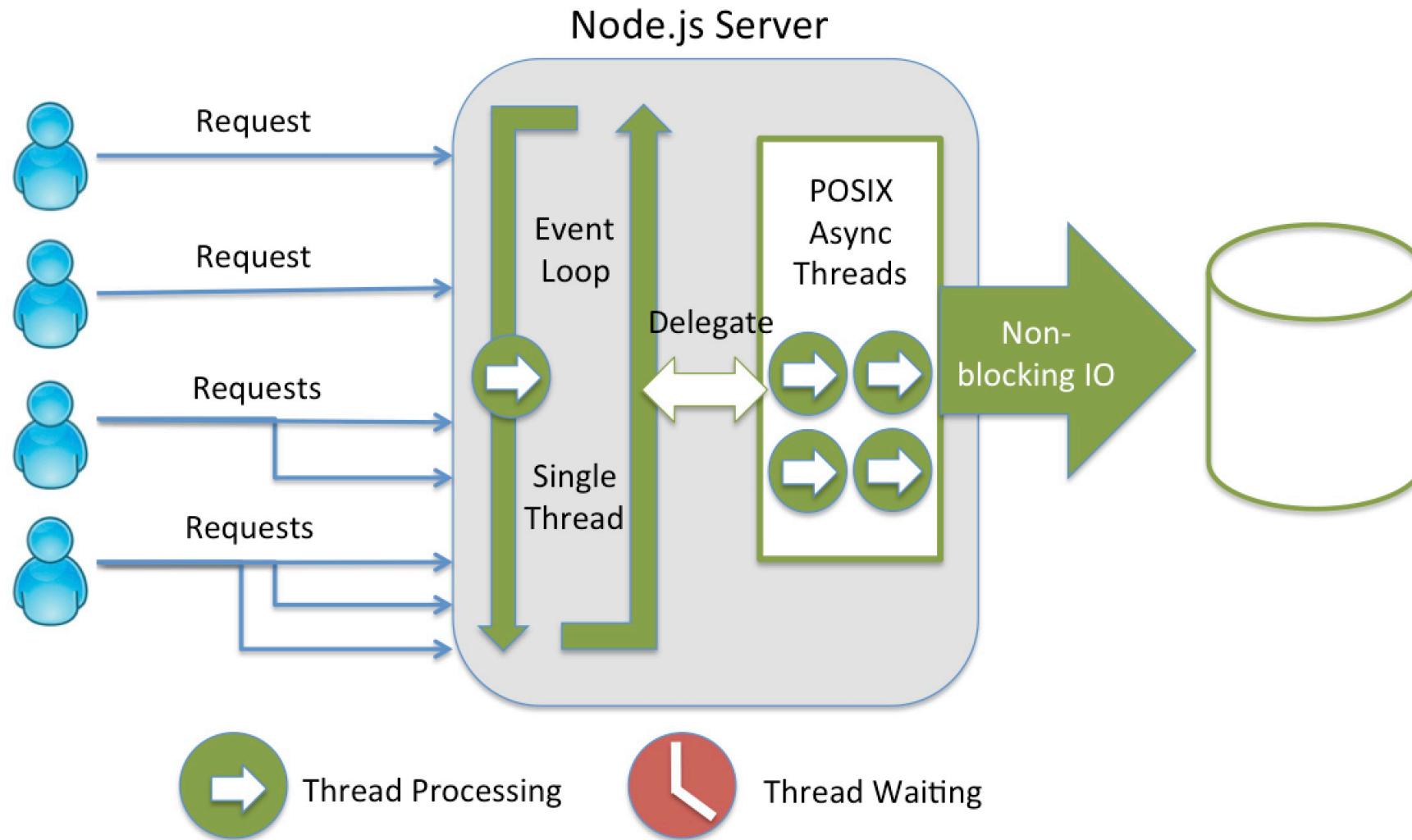
	CPU Time	System Time	RAM
PHP 5.6.4	102.69s	104.20s	2497508 KB
HHVM 3.5.0	12.56s	14.83s	362488 KB
NodeJS 0.10.35	2.64s	2.64s	92240 KB

CombSort Strict CPU Test

Ref: <http://kokizzu.blogspot.be/2015/02/numeric-combsort-benchmark-updated.html>



Kiến trúc NodeJS





2. Cài đặt môi trường



Cài đặt môi trường

- Tải ứng dụng NodeJS từ trang chủ <https://nodejs.org/en/>

The screenshot shows the official Node.js website. At the top, there's a dark header with the Node.js logo and a navigation bar with links: HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, CERTIFICATION, and NEWS. Below the header, a green banner states "Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine." Two large download buttons are prominently displayed: one for "14.17.3 LTS" (Recommended For Most Users) and another for "16.5.0 Current" (Latest Features). At the bottom, there are links for "Other Downloads", "Changelog", and "API Docs" for both LTS and Current versions.

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Download for macOS (x64)

14.17.3 LTS
Recommended For Most Users

16.5.0 Current
Latest Features

Other Downloads | Changelog | API Docs Other Downloads | Changelog | API Docs

Or have a look at the [Long Term Support \(LTS\) schedule](#).



Cài đặt môi trường

- Sau khi cài đặt, bật terminal thực hiện 2 lệnh sau để kiểm tra

```
apple > ~
> node --version
v14.17.2

apple > ~
> npm --version
7.19.1
```

The image shows a dark-themed macOS terminal window. It contains two separate command-line sessions. The first session starts with 'apple > ~' followed by the command 'node --version' and its output 'v14.17.2'. The second session starts with 'apple > ~' followed by the command 'npm --version' and its output '7.19.1'. Each session has a small green 'system' icon with a house symbol in the top right corner.



3. Function Expression



Function Expression

```
js index.js > ...
1  var say = function(message) {
2      console.log(`Hello, your message is '${message}'`);
3  }
4
5  say('Hello world');
6
7  function callFunction(message, func) {
8      func(message);
9  }
10
11 callFunction('Good bye', say);
12
13 function myFunction(name, callback) {
14     var result = `My name is ${name}`;
15     callback(result);
16 }
17
18 myFunction('Teacher', say);
```



```
> node index.js
Hello, your message is 'Hello world'
Hello, your message is 'Good bye'
Hello, your message is 'My name is Teacher'
```



4. Module Patterns



Module Patterns

```
JS myFunc.js > ...
1 var add = function (x, y) {
2     return (x + y);
3 }
4
5 module.exports = add;
```



```
JS main.js > ...
1 var add = require('./myFunc');
2 console.log(add(5, 4));
```



```
> node main.js
9
```



Module Patterns

```
JS myFunc.js > ...
```

```
1 const calculator = {
2     add : function (x, y) {
3         return (parseFloat(x) * 10 + parseFloat(y) * 10) / 10
4     },
5     pi : 3.14
6 }
7
8 module.exports = calculator;
```



```
> node main.js
7.140000000000001
```

```
JS main.js > ...
```

```
1 const cal = require('./myFunc');
2 console.log(cal.add(cal.pi, 4));
```



5. Xử lý File với NodeJS



Đọc file

```
JS file.js > ...
1 const fs = require('fs');
2 const filePath = __dirname + '/data/data.json';
3
4 var contents = fs.readFileSync(filePath, 'utf-8');
5 console.log(contents);
6 console.log('Hoàn thành đọc file');
```



```
> node file.js
[ {
    "id": 1,
    "name": "iPhone 12",
    "price": 1000
},
{
    "id": 2,
    "name": "iPad pro M1",
    "price": 1200
},
{
    "id": 4,
    "name": "macBook pro M1",
    "price": 2000
},
{
    "id": 3,
    "name": "ABC",
    "price": 1200
},
{
    "id": 5,
    "name": "FEG",
    "price": 2222
}]
```

Hoàn thành đọc file

data > {} data.json > ...

```
1 [ {
2     "id": 1,
3     "name": "iPhone 12",
4     "price": 1000
5 },
6 { "id": 2,
7     "name": "iPad pro M1",
8     "price": 1200
9 },
10 { "id": 4,
11     "name": "macBook pro M1",
12     "price": 2000
13 },
14 { "id": 3,
15     "name": "ABC",
16     "price": 1200
17 },
18 { "id": 5,
19     "name": "FEG",
20     "price": 2222
21 },
22 { "id": 6,
23     "name": "FEG",
24     "price": 2222
25 },
26 { "id": 7,
27     "name": "FEG",
28     "price": 2222
29 }]
```



Đọc file

```
file.js > ...
1  const fs = require('fs');
2  const filePath = __dirname + '/data/data.json';
3
4  fs.readFile(filePath, 'utf-8', function(err, contents) {
5      if (err != null)
6          console.log('Lỗi đọc file');
7      else
8          console.log(contents);
9  });
10
11 console.log("Hoàn thành đọc file");
```



```
> node file.js
Hoàn thành đọc file
[
  {
    "id": 1,
    "name": "iPhone 12",
    "price": 1000
  },
  {
    "id": 2,
    "name": "iPad pro M1",
    "price": 1200
  },
  {
    "id": 4,
    "name": "macBook pro M1",
    "price": 2000
  },
  {
    "id": 3,
    "name": "ABC",
    "price": 1200
  },
  {
    "id": 5,
    "name": "FEG",
    "price": 2222
  }
]
```

```
data > {} data.json > ...
1  [
2    {
3      "id": 1,
4      "name": "iPhone 12",
5      "price": 1000
6    },
7    {
8      "id": 2,
9      "name": "iPad pro M1",
10     "price": 1200
11   },
12   {
13     "id": 4,
14     "name": "macBook pro M1",
15     "price": 2000
16   },
17   {
18     "id": 3,
19     "name": "ABC",
20     "price": 1200
21   },
22   {
23     "id": 5,
24     "name": "FEG",
25     "price": 2222
26   }
27 ]
```



6. Xử lý Folder với NodeJS



Xử lý Folder với NodeJS

```
JS file.js > ...
1  const fs = ...
2
3  var folderName = 'newFolder';
4  fs.mkdir(folderName);
5
6  fs.readdirSync('./newFolder')
7    .forEach(function(file) {
8      fs.unlink(`./${folderName}/${file}`);
9      console.log(`Delete ${file}`);
10     })
11
12 fs.rmdirSync(`./${folderName}`);
```



7. Dựng Web-server với NodeJS



Dựng Web-server với NodeJS

- Các thông số cần lưu ý
 - Domain/IP Host là địa chỉ truy cập vào ứng dụng Web
 - Port là cổng truy cập vào ứng dụng Web
- Thông thường đối với ứng dụng NodeJS
 - Host = localhost / 127.0.0.1
 - Port = 3000



Dựng Web-server với NodeJS

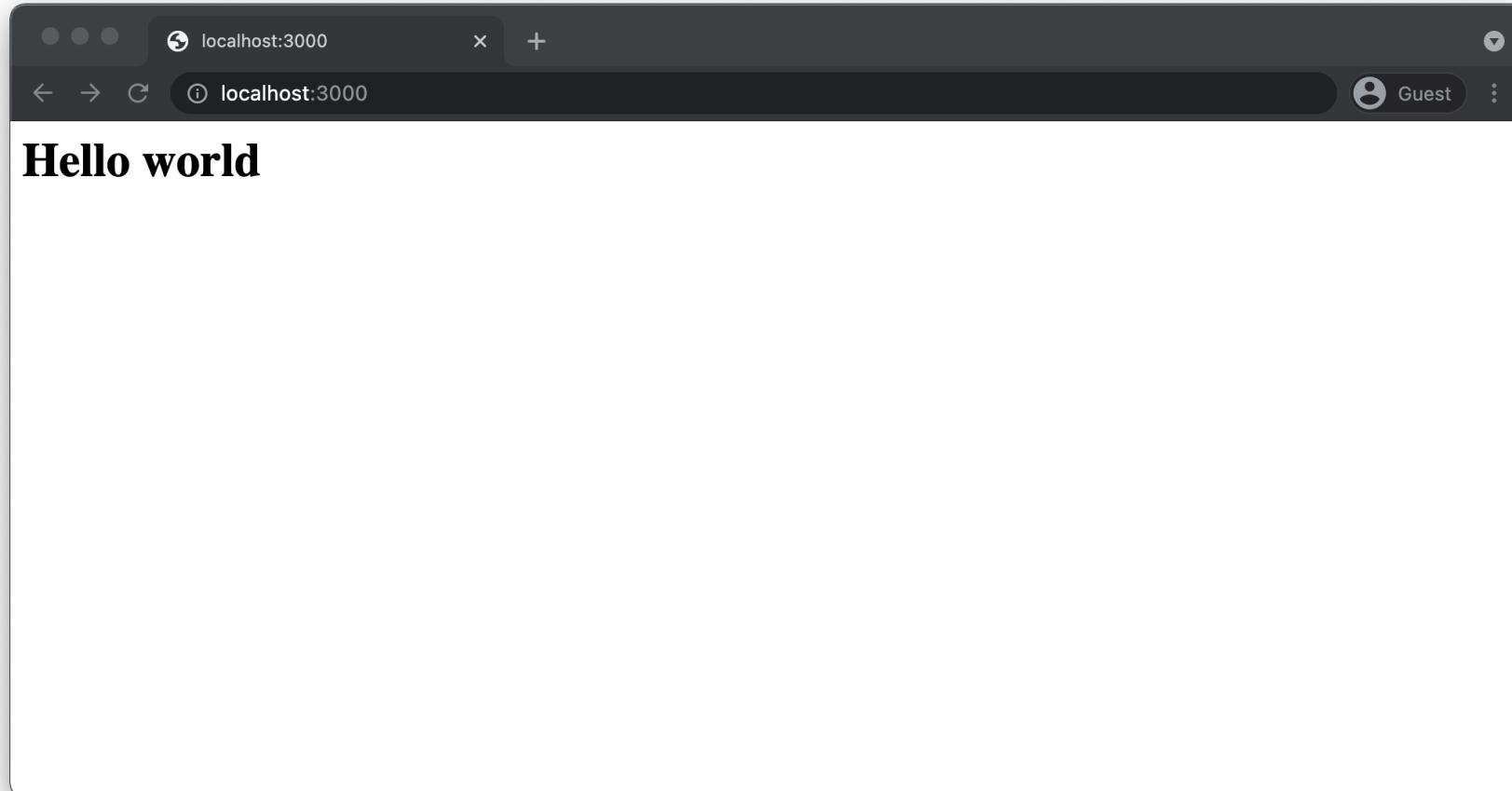
```
JS server.js > ...
1  'use strict'
2
3  var http = require('http')
4  var url = require('url')
5
6  const hostname = 'localhost' // '127.0.0.1'
7  const port = 3000
8
9  // setup server
10 const server = http.createServer((req, res) => {
11   res.statusCode = 200
12   res.setHeader('Content-Type', 'text/html')
13   res.end("<h1>Hello world</h1>")
14 })
15
16 server.listen(port, hostname, () => {
17   console.log(`Server is running at http://\${hostname}:\${port}`)
18 })
```



```
> node server.js
Server is running at http://localhost:3000
```



Truy cập bằng Web Browser





Sử dụng phần mềm Postman

The screenshot displays two main sections of the Postman website. On the left, a white box highlights the 'The Postman app' section, which describes the app's features and provides a large orange 'Download the App' button. Below this, there are links for 'Version 8.8.0 | Release Notes | Product Roadmap' and download links for Windows and Linux. On the right, a larger dark-themed window shows the Postman application interface. It has a sidebar with collections, APIs, environments, mock servers, monitors, and history. The main area shows a request for 'Twitter API v2 / Tweet Lookup / Single Tweet' with a 'GET' method and URL 'https://api.twitter.com/2/tweets/:id'. The 'Params' tab is selected, showing a table with a single row for 'id' with value '1403216129661628420'. The 'Body' tab shows a JSON response with one object containing 'data': { 'id': '1403216129661628420', 'text': 'Donovan Mitchell went down after a collision with Paul George toward the end of Game 2. https://t.co/Y9ihXhDLDN' }. The bottom of the interface includes tabs for Body, Cookies, Headers, Test Results, and a status bar indicating '200 OK 468 ms 734 B Save Response'.

<https://www.postman.com/downloads/>



Sử dụng phần mềm Postman

The screenshot shows the Postman application window. On the left, there's a sidebar with icons for Collections, APIs, Environments, Mock Servers, Monitors, and History. A central message says "You don't have any collections" with a "Create Collection" button. The main workspace has a title bar "localhost:3000" with a "GET" method selected. Below it, the "Params" tab is active, showing a table with one row: "Key" (Key) and "Value" (Value). The "Body" tab is also visible, showing the response body: "`<h1>Hello world</h1>`". The status bar at the bottom indicates "Status: 200 OK Time: 17 ms Size: 168 B".



Sử dụng phần mềm Insomnia

The screenshot shows the Insomnia website with a dark theme. At the top, there is a navigation bar with the Insomnia logo, a star rating of 16,975, and links for Products, Docs, Pricing, Plugins, Login, and a prominent "Get Started for Free" button.

The main content area features a large banner with the text "Build APIs that work." and a subtext: "Deliver high quality APIs through standards and collaboration with the Insomnia API design platform." Below this is another "Get Started for Free" button and a "Explore Plugins" link.

A central panel displays the "Swagger Petstore" API documentation. It includes a code editor showing the OpenAPI specification for the Petstore API, a detailed description of the API, and a list of operations grouped under the "pet" resource. Each operation is shown with its HTTP method, URL, and a brief description.

At the bottom of the page, there are logos for Netflix, logDNA, 1800 contacts, KAYAK, box, and cisco.

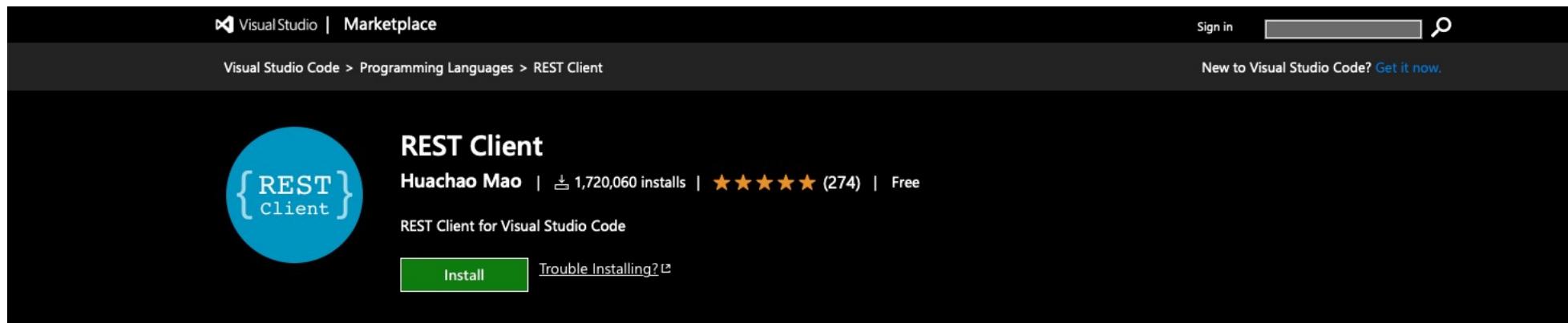


Sử dụng phần mềm Insomnia

The screenshot shows the Insomnia API client interface. At the top, it displays "Insomnia – localhost:3000". The main header includes "Dashboard / Insomnia", "No Environment", "Cookies", and a "Send" button. To the right of the "Send" button are status indicators: "200 OK", "8.03 ms", and "20 B". Below the header, there's a dropdown menu for "Body" and tabs for "Auth", "Query", "Header", and "Docs". On the right side of the interface, the response body is displayed with the text "Hello world". A small icon of a hand pointing up is centered below the response body, with the text "Select a body type from above" underneath it.



Sử dụng extention REST Client trong Visual Code





Bài tập về nhà

- Xây dựng một ứng dụng NodeJS thực hiện các yêu cầu sau:
 - Xây dựng ứng dụng Web-Server thực hiện các phép tính với các số Nguyên:
 - Phép cộng (2 điểm)
 - Phép trừ (2 điểm)
 - Phép nhân (2 điểm)
 - Phép chia (2 điểm)
 - Dựng được web-server (2 điểm)

