



# Lập trình ứng dụng web bằng NodeJS

## Bài 2. *Tạo server bằng thư viện Http*

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





# Nội dung

---

1. Http trong NodejS
2. Tạo server bằng Http trong NodeJS
3. Request và Reponse



# 1. Http trong NodeJS

---

- ❑ Http là một interface trong NodeJS được thiết kế để hỗ trợ các chức năng của giao thức Http, giúp lập trình viên làm việc dễ dàng hơn.
- ❑ Và trong NodeJS chúng ta sử dụng Http để tạo ra server giả lập lắng nghe từ giao thức Http và cổng giao thức sẽ do lập trình viên đặt ra.



# Nội dung

---

1. Http trong NodejS
2. Tạo server bằng Http trong NodeJS
3. Request và Response



# Tạo Server bằng Http trong NodeJS

- ❑ Để có thể sử dụng Http chúng ta phải chèn vào trang như sau:

```
http = require("http");
```

- ❑ Tạo ra một server

```
var http = require('http');  
  
var server = http.createServer(function(request, response) {  
    //code xử lý  
});
```



# Nội dung

---

1. Http trong NodejS
2. Tạo server bằng Http trong NodeJS
3. Request và Response



### 3. Request và Response

---

- ❑ Request là yêu cầu từ phía client gửi lên server.
- ❑ Và ở Code tạo server chúng ta thấy rất rõ đối tượng Request đang được truyền vào một biến để tiện cho việc xử lý
- ❑ Trong Request được gửi vào NodeJS có một số thuộc tính quan trọng cần phải nhớ để tiện cho việc viết website như sau: headers, method, url, body,...



### 3. Request và Response

---

- ❑ Response là việc trả lời lại client sau khi xử lý từ yêu cầu của Request.
- ❑ Tương tự như Request, đối tượng Response cũng được truyền vào một biến để tiện xử lý
- ❑ Trong Response có các phương thức quan trọng như sau: `send`, `end`, `json`,...





### 3. Request và Response

- ❑ Lệnh xem thuộc tính headers của request:

```
var headers = request.headers;  
console.log(headers);
```

- ❑ Kết quả:

```
{ host: 'localhost:8080',  
  connection: 'keep-alive',  
  'cache-control': 'max-age=0',  
  'upgrade-insecure-requests': '1',  
  'user-agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36',  
  'accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8',  
  'accept-encoding': 'gzip, deflate, sdch, br',  
  'accept-language': 'vi-VN,vi;q=0.8,fr-FR;q=0.6,fr;q=0.4,en-US;q=0.2,en;q=0.2',  
  cookie: '__atuvc=18%7C3' }
```



### 3. Request và response

---

#### ❑ Lệnh xem thuộc tính method và url:

```
var method = request.method;  
console.log("Phương thức " + method);  
var url = request.url;  
console.log("Đường dẫn: " + url);
```

#### ❑ Kết quả:

```
Phương thức GET  
Đường dẫn: /
```



### 3. Request và response

---

❑ In dữ liệu ra màn hình website cách đơn giản nhất trong NodeJS là dùng phương thức end của đối tượng Response.

❑ Cấu trúc lệnh như sau:

`response.end([data][, encoding][, callback])`

- Data: là dữ liệu truyền vào xuất ra màn hình, có thể là kiểu chuỗi hay một buffer
- Encoding: kiểu mã hóa chuỗi xuất ra màn hình
- Callback: là phương thức sẽ được gọi sau khi câu lệnh xuất ra màn hình hoàn thành

# Thảo luận

---

