



Lập trình ứng dụng web bằng NodeJS

Bài 3. *Module và Events*

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





Nội dung

1. Module là gì?

2. Cấu trúc Module trong NodeJS

3. Gọi, sử dụng module trong NodeJS

4. Events

- Event là gì?
- Cách tạo Event
- Cách gọi Event
- Event trong Module



1. Module là gì?

- ❑ Module là một thành phần trong một ứng dụng lớn, bản thân nó có thể tự hoạt động một chức năng riêng.
- ❑ Mỗi module chứa tập hợp các hàm chức năng cho cùng một đối tượng hay một mục đích chung.
- ❑ Có 2 loại Module trong NodeJS:
 - Module core, là module được NodeJS cung cấp, xây dựng sẵn
 - Module do developer xây dựng nên.



Nội dung

1. Module là gì?

2. Cấu trúc Module trong NodeJS

3. Gọi, sử dụng module trong NodeJS

4. Events

- Event là gì?
- Cách tạo Event
- Cách gọi Event
- Event trong Module



2. Cấu trúc module trong NodeJS

❑ Cấu trúc cơ bản của một module trong NodeJS:

```
exports.tên_hàm = function(danh_sách_tham_số){  
    // code xử lý  
}
```

❑ Một ví dụ cụ thể:

- Xây dựng module chào:

```
exports.ham_chao = function(){  
    console.log("Xin chào các bạn đã tạo module trong NodeJS")  
};
```



Nội dung

1. Module là gì?
2. Cấu trúc Module trong NodeJS
3. Gọi, sử dụng module trong NodeJS
4. Events
 - Event là gì?
 - Cách tạo Event
 - Cách gọi Event
 - Event trong Module



3. Gọi, sử dụng Module trong NodeJS

- ❑ Để có thể sử dụng module trong NodeJS, đầu tiên phải chèn module vào trong trang chính cần sử dụng với lệnh “require”, cấu trúc như sau:

```
tên_biến = require("đường_dẫn_đến_module");
```

- ❑ Ví dụ cụ thể với module chào:

```
module_chao = require("../modules/module_chao.js");
```



3. Gọi, sử dụng Module trong NodeJS

- ❑ Hoàn thành ví dụ sử dụng module đơn giản ở trang main.js:

```
http = require("http");
module_chao = require("../modules/module_chao.js");

http.createServer(function(request, response) {
    module_chao.ham_chao();
}).listen(8080);
```




Nội dung

1. Module là gì?
2. Cấu trúc Module trong NodeJS
3. Gọi, sử dụng module trong NodeJS
4. Events
 - Event là gì?
 - Cách tạo Event
 - Cách gọi Event
 - Event trong Module



4.1 Events

❑ Events là gì?

- Trong Javascript events là sự kiện chịu tác động từ người dùng lên HTML DOM ELEMENT, sẽ chọn một hàm để xử lý.
- Phần lớn hệ thống NodeJS được xây dựng trên kiến trúc hướng sự kiện, trong đó, các loại đối tượng định kỳ phát ra các sự kiện được gọi là emitters.
- Các Emitters sẽ gọi đến các phương thức, chức năng đang lắng nghe được gọi là listener



Nội dung

1. Module là gì?
2. Cấu trúc Module trong NodeJS
3. Gọi, sử dụng module trong NodeJS
4. **Events**
 - Event là gì?
 - Cách tạo Event
 - Cách gọi Event
 - Event trong Module



4.2 Cách tạo Event

❑ Gọi thư viện để tạo Event (EventEmitter):

- Câu lệnh:

tên_class = require("events").EventEmitter;

tên_biến = new EventEmitter();

- Ví dụ:

```
EventEmitter = require("events").EventEmitter;  
my_event = new EventEmitter();
```



4.2 Cách tạo Event

❑ Cách tạo một Listener trong NodeJS:

- Câu lệnh:

```
Tên_biến.on("tên_sự_kiện", function(){  
    //code xử lý  
});
```

- Ví dụ:

```
my_event.on("lang_nghe", function () {  
    console.log("sự kiện đã được gọi");  
});
```



Nội dung

1. Module là gì?

2. Cấu trúc Module trong NodeJS

3. Gọi, sử dụng module trong NodeJS

4. Events

- Event là gì?
- Cách tạo Event
- Cách gọi Event
- Event trong Module



4.3 Cách gọi event

❑ Để gọi event chạy thì EventEmitter có câu lệnh như sau:

- Cấu trúc:

- Tên_biến.emit("tên_sự_kiện");

- Ví dụ:

```
my_event.emit("lang_nghe");
```



Nội dung

1. Module là gì?
2. Cấu trúc Module trong NodeJS
3. Gọi, sử dụng module trong NodeJS
4. Events
 - Event là gì?
 - Cách tạo Event
 - Cách gọi Event
 - Event trong Module



4.4 Event trong module

- ❑ Event trong module có nghĩa là xây dựng một module có các chức năng được gọi từ các event xây dựng bên trong một cách dễ dàng. Và ví dụ dễ thấy nhất của chúng ta chính là áp dụng event của thư viện http như sau:

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

var server = http.createServer();

server.on("request", function (req, res) {
    res.end("this is the response");
});

server.listen(8080);
```

Thảo luận

