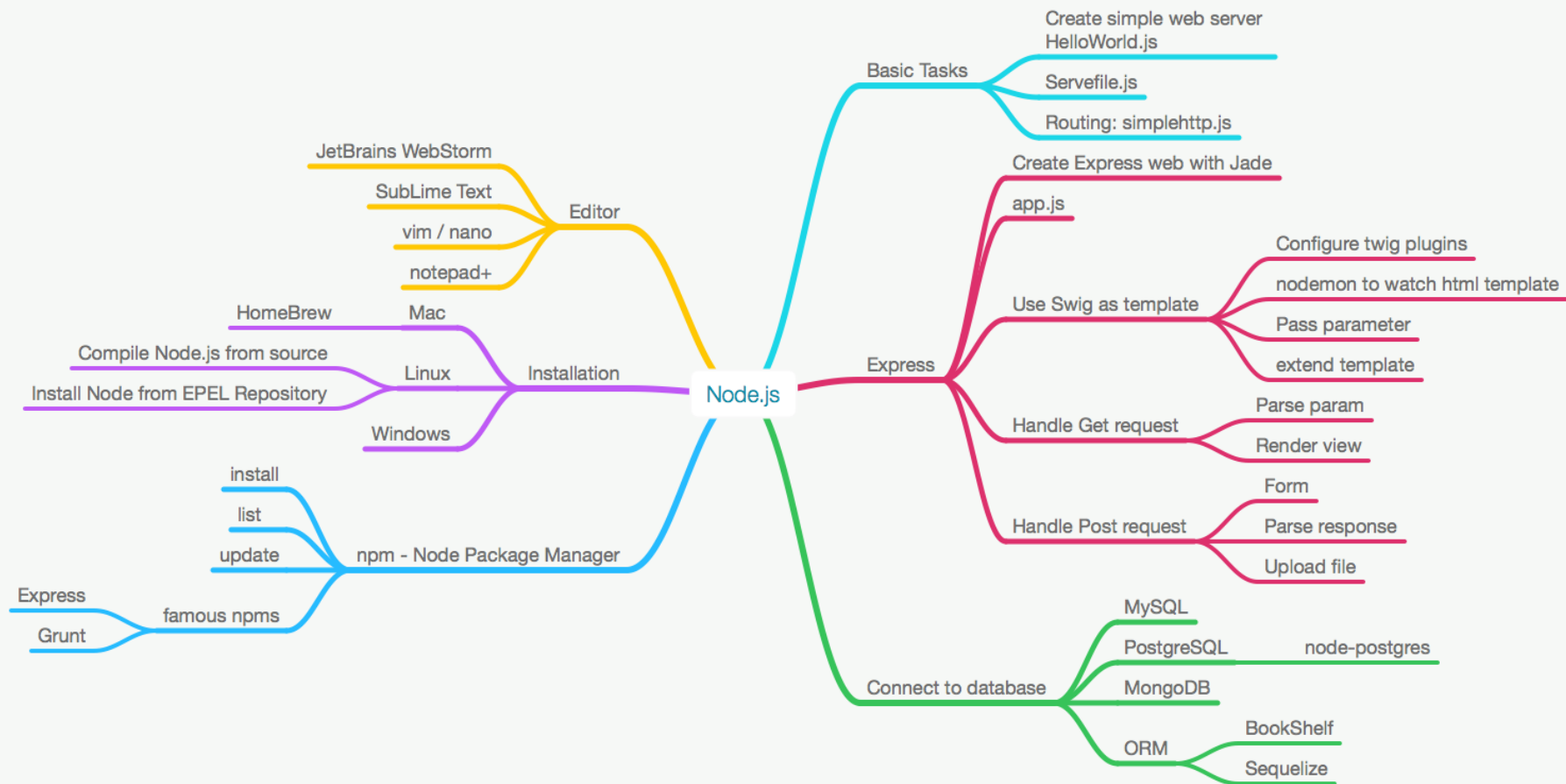


# Node kết hợp với Express

thanhnv@techmaster.vn



# Giới thiệu Express

---

- Express là một framework của nodejs
- Dễ dàng tạo các REST service
- Code javascript thông dụng
- Hỗ trợ nhiều template engine (jade, ejs, swig...)

# Cài đặt Express

- Cài đặt Express bằng lệnh

`npm install express`

- Cài đặt module express generator để gen project express

`sudo npm install -g express-generator`

- Khởi tạo project express bằng lệnh

`express project_name`

# Tạo project 'music'

---

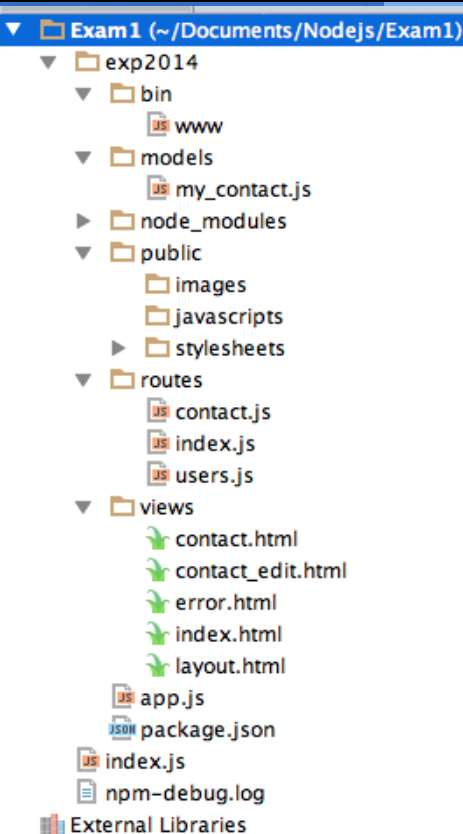
```
$ express music
```

```
$ cd music
```

```
$ npm install
```

```
$ DEBUG=music ./bin/www
```

# Cấu trúc thư mục ứng dụng Express



- Để chạy ứng dụng express dùng lệnh

**DEBUG=project\_name ./bin/www**

*Nhớ có space giữa project\_name và ./bin/www*

*Có thể gõ lệnh **npm start***

- File bin/www định nghĩa ban đầu của ứng dụng như tên ứng dụng, cổng để chạy ứng dụng

# File bin/www

```
#!/usr/bin/env node
var debug = require('debug')('helloworld');
var app = require('../app');

app.set('port', process.env.PORT || 3000);

var server = app.listen(app.get('port'), function() {
  debug('Express server listening on port ' + server.address().port);
});
```

# Cài đặt Nodemon

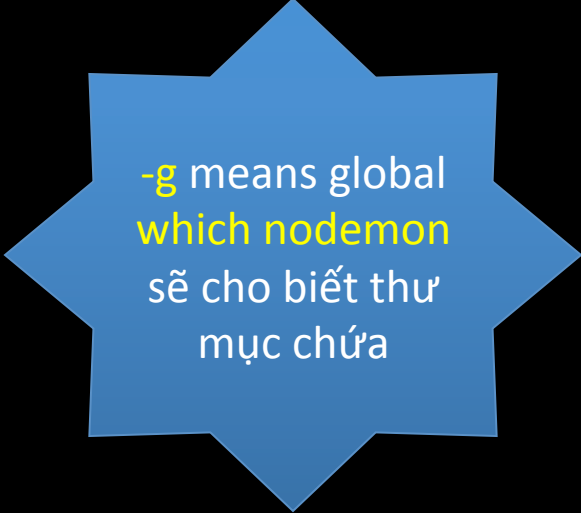
Nodemon là một module giúp cho ứng dụng có thể cập nhật code ngay khi lập trình viên save file

`npm install -g nodemon`

Kiểm tra xem nodemon đã cài chưa

`npm list -g nodemon`

`which nodemon`



-g means global  
which nodemon  
sẽ cho biết thư  
mục chứa



# Chạy ứng dụng với Nodemon

## Nodemon bin/www

```
^Ccuongs-Mac-Pro:coolsock techmaster$ Nodemon bin/www
11 Nov 22:08:24 - [nodemon] v1.2.1
11 Nov 22:08:24 - [nodemon] to restart at any time, enter `rs`
11 Nov 22:08:24 - [nodemon] watching: *.*
11 Nov 22:08:24 - [nodemon] starting `node bin/www`
GET / 304 235.509 ms - -
GET /stylesheets/style.css 304 2.183 ms - -
GET / 200 13.067 ms - 185
GET /stylesheets/style.css 304 0.675 ms - -
```

# Tạo REST service đơn giản

---

- Tạo một project express
- Xử lý routes để quy định trả dữ liệu json
- Tham khảo <http://expressjs.com/api.html>

```
var express = require('express');  
var router = express.Router();
```



Sửa file routes/index.js

```
/* GET home page. */  
router.get('/', function(req, res) {  
  res.render('index', { title: 'Techmaster.vn' });  
});  
  
router.get('/songs', function(req, res) {  
  res.set('Content-Type', 'application/json');  
  res.send({"songs": [  
    {"name": "Fix You", "singer": "Cold Play"},  
    {"name": "Blank Space", "singer": "Taylor Swift"},  
    {"name": "Because of You", "singer": "Kelly Clarkson"}  
  ]  
});  
});
```

JSON

[« Home](#) | [Reset filters](#)

Clear

☐ Apps☒ Extensions☐ Themes

CATEGORIES



## JSON Formatter

from callumlocke.co.uk

Makes JSON easy to read. Open source.

+ FREE

Developer Tools

★★★★★ (443)

ADDED



## JSON Viewer

tulios

The most beautiful and customizable JSON/JSONP highlighter

★ RATE IT

Developer Tools

★★★★★ (65)

```
localhost:3000/songs
localhost:3000/songs

// 20141228123140
// http://localhost:3000/songs

{
  "songs": [
    {
      "name": "Fix You",
      "singer": "Cold Play"
    },
    {
      "name": "Blank Space",
      "singer": "Taylor Swift"
    },
    {
      "name": "Because of You",
      "singer": "Kelly Clarkson"
    }
  ]
}
```



# Dùng template trong Express

# Có nhiều loại template

---

- Jade
- HandleBars
- Swig

# Jade

```
router.get('/', function(req, res) {  
  res.render('index', { title: 'Techmaster.vn' });  
});
```

```
extends layout
```

```
block content
```

```
h1= title
```

```
p Welcome to #{title}
```

```

router.get('/mysongs', function(req, res) {
  res.render('mysongs', { title: 'My favorite song', songs: [
    {"name": "Fix You", "singer": "Cold Play"},
    {"name": "Blank Space", "singer": "Taylor Swift"},
    {"name": "Because of You", "singer": "Kelly Clarkson"}
  ]});
});

```

```

extends layout
block content
  h1= title
  p Welcome to #{title}
  table
    each song, i in songs
      tr
        td #{i}
        td #{song.name}
        td #{song.singer}

```

# My favorite song

Welcome to My favorite song

0 Fix You	Cold Play
1 Blank Space	Taylor Swift
2 Because of You	Kelly Clarkson



# Swig hay hơn Jade

- Jade hơi khó hiểu với dev đã quen HTML
- <http://george.mand.is/2014/02/chose-swig-jade-templates-nodeexpress-project/>
- <http://paularmstrong.github.io/node-templates/benchmarks.html>

# Sử dụng swig trong Express

- Cài đặt swig module bằng lệnh
- `npm install swig --save`
- Cài đặt consolidate module
- `npm install consolidate`
- Sửa code trong file app.js
- Tạo các file view với đuôi là html
- Cú pháp template trong file swig giống với cú pháp file volt trong phalcon

Swig has nearly identical syntax to Twig, so we can assume they can be used interchangeably speaking of syntax highlighting and code completion.

## Install Twig plugin

Use the .twig extension on files or adding a new mapping in the File Types system menu for the type Twig, with wildcard \*.html or \*.swig, how you prefer.

I'm using the \*.html workaround and work like a charm, both Swig & HTML syntax highlighting + code completion.

## Appearance &amp; Behavior

## Editor

General

Colors &amp; Fonts

Code Style

Inspections

File and Code Templates

File Encodings

Live Templates

File Types

Emmet

Images

Intentions

Language Injections

Spelling

TextMate Bundles

TODO

## Plugins

Version Control

Project: song

Build, Execution, Deployment

## Languages &amp; Frameworks

JavaScript

Schemas and DTDs

Compass

Dart

Node.js and NPM

Template Data Languages

XSLT

XSLT File Associations

## Editor &gt; File Types

## Recognized File Types

SASS files  
SCSS files  
SourceMap files  
SQL files (syntax highlighting only)  
Stylus files  
Text files  
Twig  
TypeScript files  
XHTML files  
XML Document Type Definition  
XML files  
YAML files  
YouTrack Query Language

+ -

## Registered Patterns

\*.html  
\*.twig

+ -

## Ignore files and folders

CVS;SCCS;RCS;rcs;.DS\_Store;.svn;.pyc;.pyo;\*.pyc;\*.pyo;.git;\*.hprof;.svn;.hg;\*.lib;\*.~;\_\_pycache\_\_;bundle;\*.rbc;



Cancel

Apply

OK

# Configure swig in app.js

```
var app = express();  
var swig = require('swig');  
// view engine setup  
app.engine('html', swig.renderFile);  
app.set('view engine', 'html');  
app.set('views', path.join(__dirname, 'views'));
```

Để nodemon theo dõi cả javascript và html

---

```
nodemon bin/www -e js,html
```

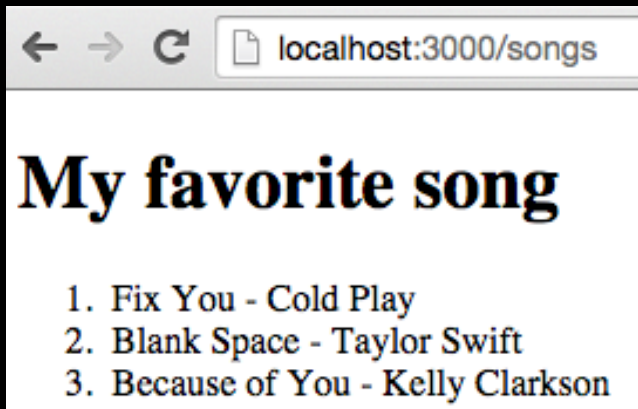
```
var express = require('express');
var router = express.Router();

/* GET home page. */
router.get('/', function(req, res) {
  res.render('index', { title: 'Express' });
});

router.get('/songs', function(req, res) {
  res.render('songs', { title: 'My favorite song', songs: [
    {"name": "Fix You", "singer": "Cold Play"},
    {"name": "Blank Space", "singer": "Taylor Swift"},
    {"name": "Because of You", "singer": "Kelly Clarkson"}
  ]});
});

module.exports = router;
```

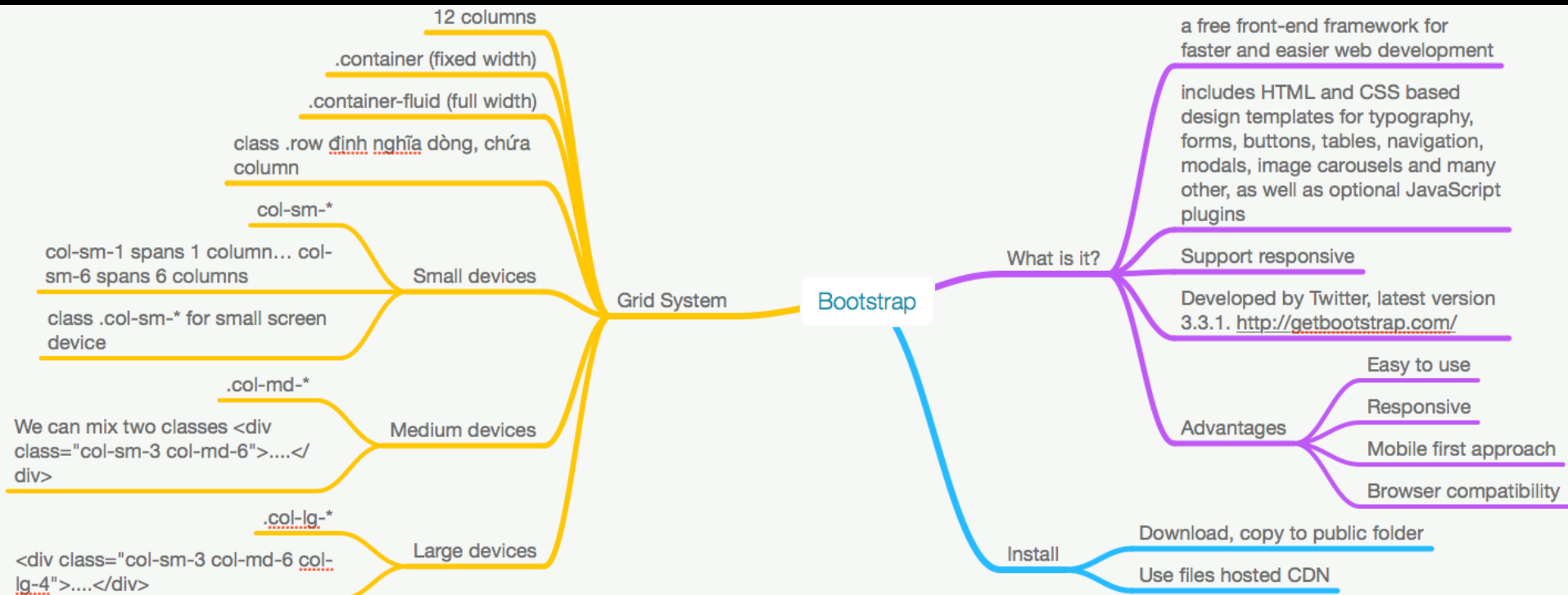
```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>{{ title }}</title>
</head>
<body>
<h1>{{ title }}</h1>
{% for song in songs %}
  {% if loop.first %}<ol>{% endif %}
  <li>{{ song.name }} - {{ song.singer }}</li>
  {% if loop.last %}</ol>{% endif %}
{% endfor %}
</body>
</html>
```







# Bootstrap



▼ song (/Volumes/DATA/notesjs/OnlineAlbum/song)

► bin

► node\_modules (library home)

▼ public

▼ bootstrap

▼ css

bootstrap.css

bootstrap.css.map

bootstrap.min.css

bootstrap-theme.css

bootstrap-theme.css.map

bootstrap-theme.min.css

▼ fonts

glyphicons-halflings-regular.eot

glyphicons-halflings-regular.svg

glyphicons-halflings-regular.ttf

glyphicons-halflings-regular.woff

▼ js

bootstrap.js

bootstrap.min.js

npm.js

images

javascripts

► stylesheets

► routes

► views

app.js

package.json

External Libraries

```
1 <!DOCTYPE html>
2 <html>
3 <head lang="en">
4   <meta charset="UTF-8">
5   {% block head %}
6   <link rel="stylesheet" href="/bootstrap/css/bootstrap.min.css">
7   <link rel="stylesheet" href="/bootstrap/css/bootstrap-theme.min.css">
8   <script src="/bootstrap/js/bootstrap.min.js"></script>
9   {% endblock %}
10  <title>{{ title }}</title>
11 </head>
12 <body>
13 <div class="container">
14   <h1>{{ title }}</h1>
15   {% block content %}{% endblock %}
16 </div>
17 </body>
18 </html>
```

# Lấy dữ liệu biến truyền vào của Request

---

- Có một số cách để có thể lấy được dữ liệu với các tham số truyền vào URL như sau

# Lấy dữ liệu biến truyền vào của Request

```
// GET /user/tj
```

```
req.params.name
```

```
// => "tj"
```

```
// GET /file/javascripts/jquery.js
```

```
req.params[o]
```

```
// => "javascripts/jquery.js"
```

# Lấy dữ liệu biến truyền vào của Request

```
// GET /search?q=tobi+ferret
```

```
req.query.q
```

```
// => "tobi ferret"
```

```
// GET /shoes?
```

```
order=desc&shoe[color]=blue&shoe[type]=converse
```

```
req.query.order
```

```
// => "desc"
```

# Lấy dữ liệu biến truyền vào của Request

```
// ?name=tobi
```

```
req.param('name')
```

```
// => "tobi"
```

```
// POST name=tobi
```

```
req.param('name')
```

```
// => "tobi"
```

# Lấy dữ liệu biến truyền vào của Request

```
// ?name=tobi
```

```
req.param('name')
```

```
// => "tobi"
```

```
// POST name=tobi
```

```
req.param('name')
```

```
// => "tobi"
```



# Lấy dữ liệu biến truyền vào của Request

---

```
// /user/tobi for /user/:name
```

```
req.param('name')
```

```
// => "tobi"
```

# Làm việc với biến Response

//Send dữ liệu về (json, plain text, html...)

```
Res.send({name: "nguyen thanh"})
```

//Render một view

```
Res.render("index",{title:"hello"})
```

//Chuyen sang một trang khác

```
Res.redirect("/hello")
```



**WebSocket server → Socket.io**

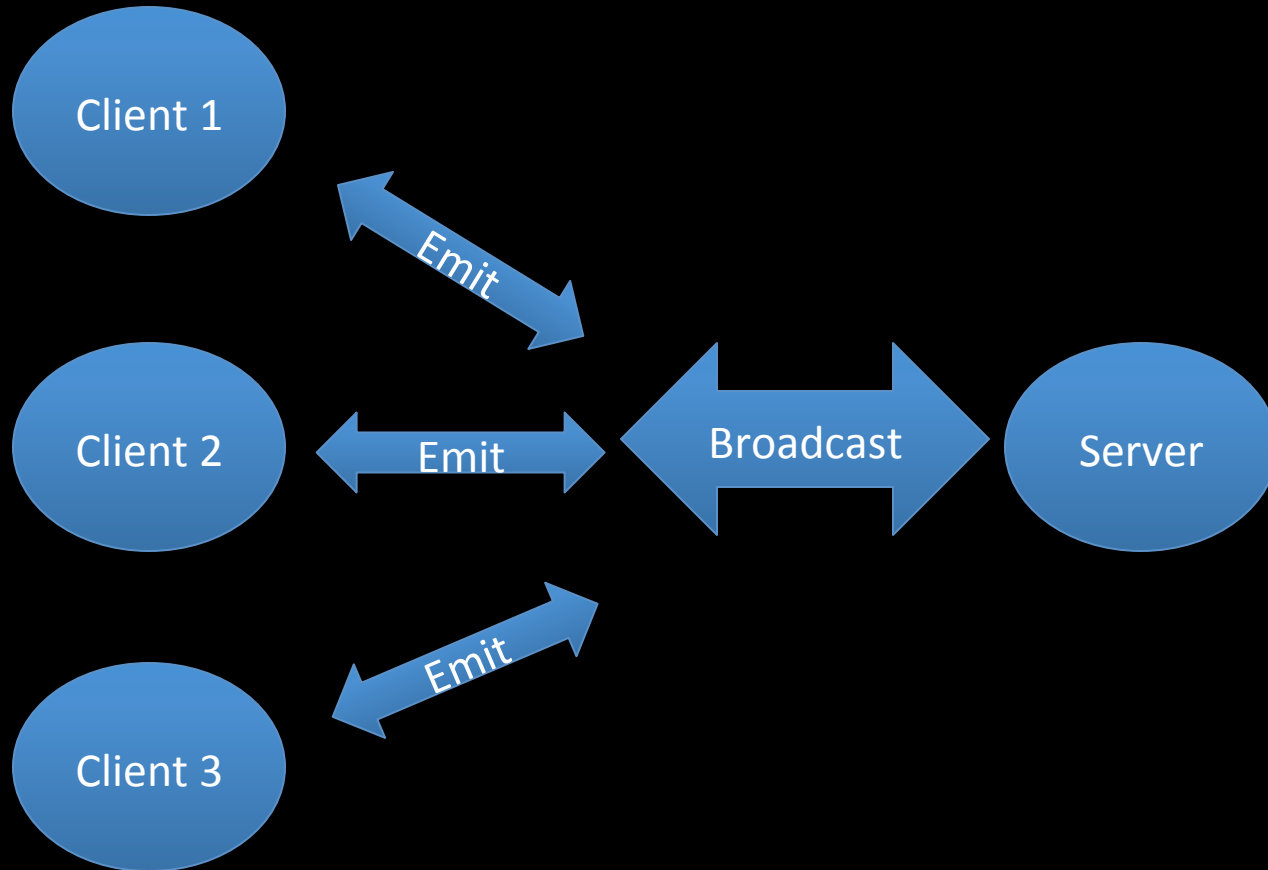
# Install Socket.io

---

Để cài đặt socket trong express chỉ cần gõ lệnh

```
npm install socket.io
```

# Socket.io



# Server trên Express

## Chỉnh sửa file bin/www.js

```
var io = require("socket.io").listen(server);  
var users = [];  
var allSockets = [];  
io.sockets.on('connection', function (socket) {  
  socket.on('connected', function (data) {  
    if (users.indexOf(data.name) < 0)  
      users.push(data.name);  
    if (allSockets.indexOf(socket) < 0) {  
      allSockets.push(socket);  
    }  
  })  
})
```

# Server trên Express

## Cài đặt sự kiện lắng nghe khi có client kết nối

```
var allSockets = [];  
io.sockets.on('connection', function (socket) {  
  socket.on('connected', function (data) {  
    if (users.indexOf(data.name) < 0)  
      users.push(data.name);  
    if (allSockets.indexOf(socket) < 0) {  
      allSockets.push(socket);  
    }  
    io.sockets.emit("connected", {  
      name: data.name,  
      users: users  
    });  
  });  
});
```

# Server trên Express

## Cài đặt một số sự kiện khác nếu muốn

```
socket.on('disconnect', function (room) {
  console.log("disconnect");
  var index = allSockets.indexOf(socket);
  if (index > -1) {
    socket.disconnect();
    allSockets.splice(index, 1);
    var name = users[index];
    console.log(name+" disconnect");
    users.splice(index, 1);
    io.sockets.emit("disconnected", {
      name: name,
      users: users
    });
  }
});

socket.on('subscribe', function (data) {
  console.log('joining room', data.room);
  socket.join(data.room);
  io.sockets.emit("invite", data);
});

socket.on('unsubscribe', function (data) {
  console.log('leaving room', data.room);
  socket.leave(data.room);
});
```



# Client trên Express

Include thư viện javascript cần thiết

```
{% block footer %}  
  <script src="socket.io/socket.io.js"></script>  
</script>
```

# Client trên Express

## Khởi tạo đối tượng socket và kết nối với Server

```
socket = io.connect('http://localhost:3000');
```

# Client trên Express

## Cài đặt socket lắng nghe các sự kiện gửi từ Server

```
socket.on("connected", function (data) {  
    if (data.name) {  
        var node = document.createElement("li");  
        var textnode = document.createTextNode('Welcome ' + data.name + ' to the chat');  
        node.appendChild(textnode);  
        content.appendChild(node);  
        var html = '';  
        for (var i = 0; i < data.users.length; i++) {  
            if (data.users[i] !== name)  
                html += '<li onclick="createPrivateRoom(this)">' + data.users[i] + '</li>';  
        }  
        listUsers.innerHTML = html;  
    }  
});
```