

Node et technologies HTML5

Florent Marchand de Kerchove
Merwan Achibet

UFR sciences et techniques
Université du Havre

9 octobre 2011

Introduction



1 Node et compagnie

Node

Express

npm

2 HTML5

Canvas HTML

WebSocket et Socket.IO

Node

Description

<http://nodejs.org/>

Serveur :

- événementiel,
- asynchrone,
- en JavaScript.

Intérêts :

- Language identique client/serveur
- Performance élevée (100 000+ connexions simultanées)
- API réseau élémentaire
- Riche librairie de modules

Similaire à Twisted ou EventMachine

Node

Serveur événementiel

Reçoit des évènements et y répond

- Semblable au DOM et à jQuery
- Fortement adapté aux applications réseau
- Approche dynamique

```
server.on('join', function(user) { join(user); });  
server.on('message', function(msg) { broadcast(msg) });  
server.on('leave', function(user) { leave(user); });
```

Node

Serveur asynchrone

- Entrées/sorties asynchrones (*epoll*, *kqueue*, ...)
- Le processus dort en attendant les évènements
- Aucun *thread*, un seul processus (*fork* possible)
- Plus simple à programmer

Inconvénient :

- Éviter les appels bloquants trop long

Node

Serveur en JavaScript

- Utilise le moteur JavaScript V8 de Google (Ecma-262 édition 5)
- Simplifie les échanges client-serveur
- Partage de code possible

Node

Installation manuelle

Dépôt officiel -> <http://github.com/joyent/node/>

```
cd  
git clone http://github.com/joyent/node.git  
cd node  
git checkout v0.4.12  
./configure  
make  
sudo make install
```

Prérequis :

```
sudo apt-get install build-essential git
```


Node

Exemples

Serveur écho

```
var net = require('net');

var server = net.createServer(function (socket) {
  socket.write("Echo server");
  socket.pipe(socket);
}).listen(1337);
```

Node

Exemples

Serveur HTTP

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

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World');
}).listen(1337);
```

```
ab -n 1000 -c 1000 http://localhost:1337
```

Node

Exemples

Serveur HTTP bloquant

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

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  setTimeout(function(){
    res.end('Hello World');
  }, 2000)
}).listen(1337);
```

```
ab -n 1000 -c 1000 http://localhost:1337
```

<http://expressjs.com/>

npm

Description

<http://npmjs.org/>
Node Package Manager

Ressources et documentation

1 Node et compagnie

Node

Express

npm

2 HTML5

Canvas HTML

WebSocket et Socket.IO

Canvas HTML

Description

Élément `<canvas>` :

- Permet de dessiner et d'animer librement sur une page web
- Contextes 2d et 3d (WebGL)
- API stable
- Implémenté dans les navigateurs majeurs

Alternative aux SVG plus performante :

- Surface *bitmapped* plutôt que vectorielle
- Pas d'insertion dans le DOM
- Accélération matérielle possible

Canvas HTML

Utilisation

Documentation :

<http://developer.mozilla.org/en/HTML/Canvas>

Élément HTML

```
<canvas id="canvas" width="300" height="300">  
  Canvas non supporté par le navigateur  
</canvas>
```

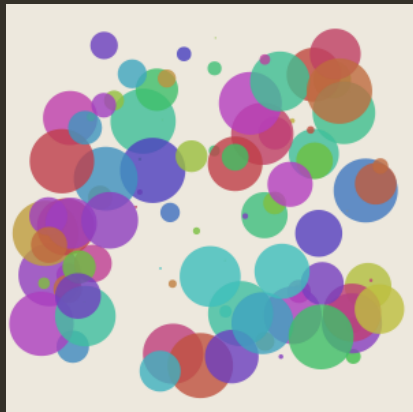
JavaScript

```
var canvas = document.querySelector('#canvas');  
var ctx = canvas.getContext('2d');
```

Canvas HTML

Exemple

```
ctx.fillStyle = 'hsl(40, 30%, 90%)';  
ctx.fillRect(0, 0, 300, 300);  
  
for (var i=0; i < 100; ++i) {  
  ctx.save();  
  ctx.translate(Math.random()*250,  
                Math.random()*250);  
  ctx.fillStyle = 'hsla(' +  
    Math.random()*360 +  
    ', 50%, 50%, 0.8)';  
  ctx.beginPath();  
  ctx.arc(24, 24,  
          12 + 12*Math.sin(i),  
          Math.PI*2, false);  
  ctx.fill();  
  ctx.restore();  
}
```



Ressources et documentation