



NODEJS

BOJ Yoann

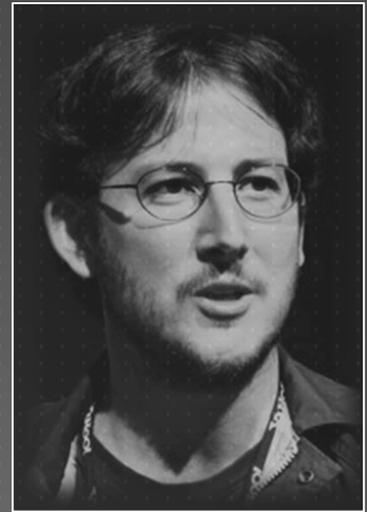
Informatique et Réseaux 3

SOMMAIRE

- ▶ Introduction
 - ▶ Les bases de NodeJS
 - ▶ Module Socket I/O
 - ▶ Performances
 - ▶ Démonstration
 - ▶ Conclusion
- 

HISTORIQUE

- ▶ NodeJS créé par Ryan Dahl en 2009
- ▶ Sponsorisé par Joyent
- ▶ Développé sur la base de la VM V8 de Google
- ▶ Codé en JavaScript



INTRODUCTION

- ▶ Asynchrone
- ▶ Non-bloquant
- ▶ Scalable
- ▶ Déclenchement d'évènements
- ▶ Beaucoup de modules disponibles
- ▶ Traitements côté serveur

LES BASES DE NODEJS

Le fameux Hello World

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

http.createServer(function (request, response) {
  response.writeHead(200, {'Content-Type': 'text/plain'});
  response.end('Hello World\n');
}).listen(80);

console.log('Adresse du serveur: http://localhost:80/');
```

TRÈS UTILE: Fonction de Callback.

DES MODULES À FOISON

- ▶ http : Serveur et client web HTTP
- ▶ net : Serveur et client TCP
- ▶ cradle : base de données en cache
- ▶ xml2js : XML vers JavaScript
- ▶ Crud-file-server : CRUD de n'importe quel type de fichier
- ▶ Djangode : framework utilisant les concepts de Django
- ▶ Mojito : MVC et librairie permettant le développement d'applications en HTML5
- ▶ Nodepress : MVC permettant le développement de blogs
- ▶ Compress : compression de données en Gzip
- ▶ Session : gestionnaire de sessions
- ▶ Form2json : formulaires vers un format Json
- ▶ Bibliothèques pour gérer tous les types de bases de données (postgre, mysql, sqlite, oracle...)

DES MODULES À FOISON

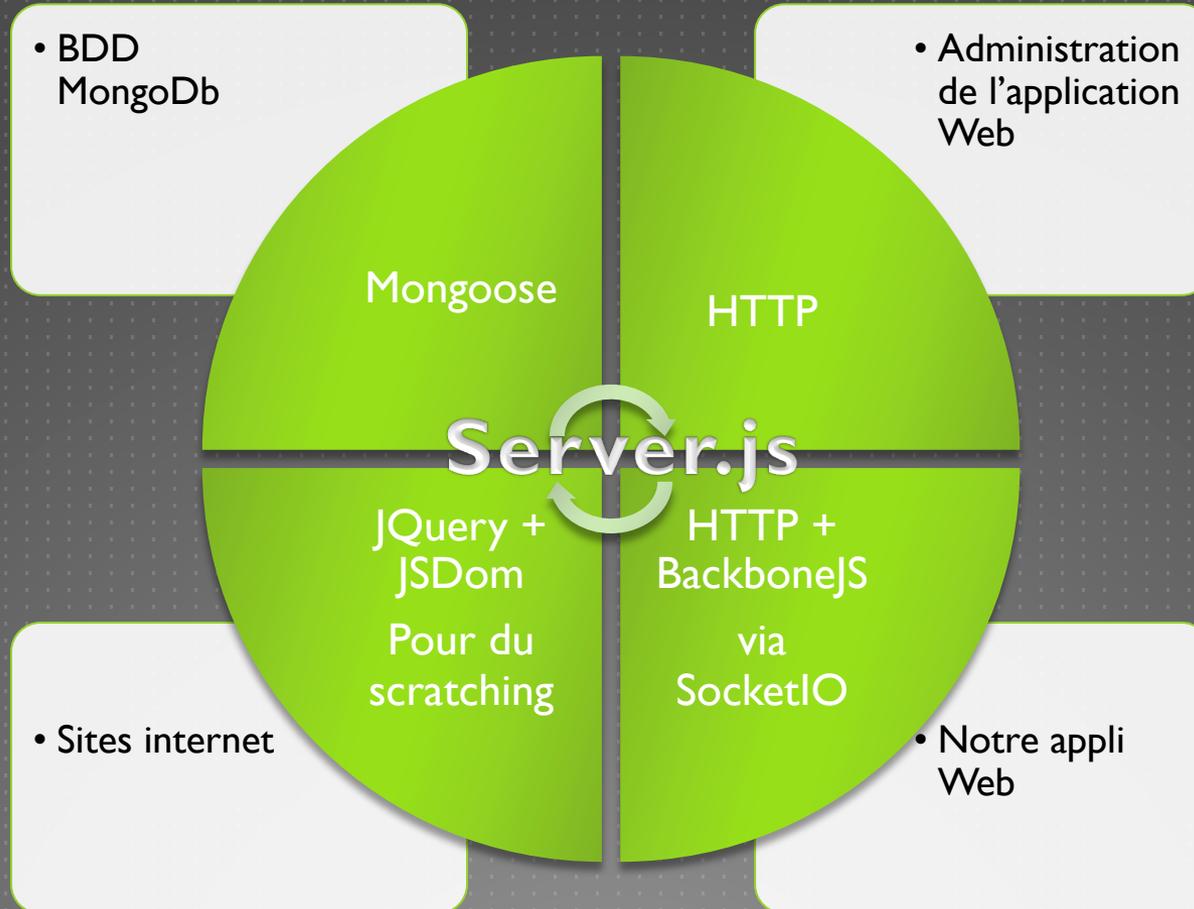
- ▶ Everyauth : connexion à plusieurs API (Facebook, GitHub, Vimeo, Yahoo, Twitter, Instagram ...)
- ▶ Emailjs : envois de mails
- ▶ Jsonjs : parser Json
- ▶ Chatio : simple chat
- ▶ Node-crontab : contrôle d'une crontab

- ▶ Possibilité de créer vos propres modules .js

NPM : Node Packaged Modules

```
$ npm install everyauth
```

DES ARCHITECTURES RICHES



MODULE SOCKET I/O

- ▶ Communication asynchrone
- ▶ Envois d'évènements entre le serveur et un ou plusieurs clients
- ▶ Broadcast ou communication avec un seul client
- ▶ Asynchrone donc pas de problème de concurrence entre les différents clients
- ▶ Utile pour les applications temps réel

MODULE SOCKET I/O

Un exemple

► Un Tchat : côté serveur

```
io.sockets.on('connection', function (socket) {  
  socket.on('chat', function (data) {  
    var address = socket.handshake.address;  
    var jsondata = {'text':data, 'name':address.address + ":" + address.port};  
    io.sockets.emit('chat',jsondata);  
  });  
});
```

MODULE SOCKET I/O

► Un Tchat : côté client

```
<script charset="utf-8" src="http://code.jquery.com/jquery-1.7.min.js"></script>
<script src="/socket.io/socket.io.js"></script>

<script>
  var socket = io.connect('http://localhost');

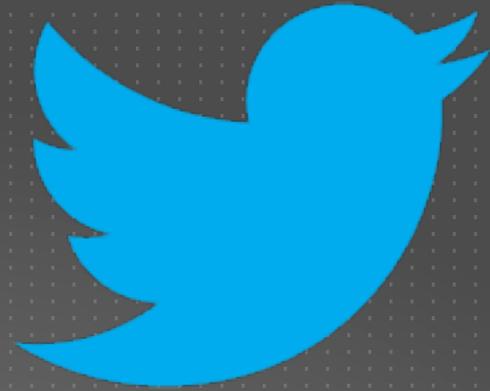
  function send(){
    var tmp = $("#texte").val();
    socket.emit("chat",tmp);
    $("#texte").val('');
  }

  socket.on('chat', function(data){
    $("#tableau").append("<tr><td>" + data['name'] + "</td><td>" + data['text'] + "</td></tr>");
  });
</script>
```

PERFORMANCES DE NODEJS

- ▶ Avec une simple utilisation, les performances sont les mêmes qu'Apache/PHP
- ▶ En mode multiserveurs, les performances sont supérieures
- ▶ Evolution avec Socket I/O très efficace, grâce à la scalability et le fonctionnement asynchrone
- ▶ NodeJS casse les barrières de langage entre le client et le serveur

DÉMONSTRATION



#MetsMoiDesA

DÉMONSTRATION

- ▶ Stream et filtre de tweets postés durant la présentation
- ▶ **Déclaration d'un API auprès de Twitter**
- ▶ **Utilisation des module ntwitter et socket I/O**

- ▶ **On peut aussi filtrer par emplacement géographique... Attention !**

DÉMONSTRATION

Côté serveur

```
var t = new twitter({
  consumer_key: credentials.consumer_key,
  consumer_secret: credentials.consumer_secret,
  access_token_key: credentials.access_token_key,
  access_token_secret: credentials.access_token_secret
});

t.stream(
  'statuses/filter',
  { track: ['#MetsMoiDesA'] },
  function(stream) {
    stream.on('data', function(tweet) {
      var tmp = {pseudo:tweet.user.screen_name, tweet:tweet.text}
      io.sockets.emit('tweet',tmp);
    });
  }
);
```

DÉMONSTRATION

Côté client

```
<script charset="utf-8" src="http://code.jquery.com/jquery-1.7.min.js"></script>
<script src="/socket.io/socket.io.js"></script>

<script>
  var socket = io.connect('http://localhost');

  socket.on('tweet',function (data) {
    $("#tableau").append("<tr><td>" + data['pseudo'] + "</td><td>" + data['tweet'] + "</td></tr>");
  });
</script>
```

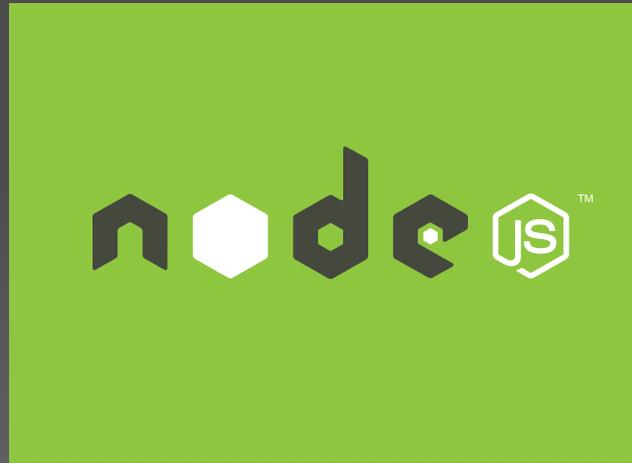
CONCLUSION

- + Sa rapidité
- + Sa modularité
- + Asynchrone

- Manque de documentation
- Modules développés par des passionnés
- Plusieurs modules pour faire la même chose

Enjoy

node.js



```
{ progression : “Fin de la présentation”,  
  text : “Avez-vous des questions ?”,  
  reminder : “#MetsMoiDesA” }
```