MyDigitalSchool
2021/2022
B3 - Développeur Web

Node.js

Mr Bonnet Denis

	contenu
03:30	Présentation du module, principes général de Node.js, environement de dev,
03:30	TD mise en place
03:30	cours : CLI commande, JS pour le back
03:30	npm, npx et modules utiles
03:30	Gestion de fichier, callback et Stream
03:30	Event emitter et listener
03:30	module http
03:30	Principe REST
03:30	TP
03:30	TP + soutenance

Node.JS (14h + 21h)

A l'issue de ce module, l'apprenant sera à même

- Fonctionnement général
- Le langage ECMAScript
- Les bibliothèques via NPM
- Réalisation d'une application simple

Table de cours

Node.js
Généralités

- Plateforme Javascript Côté serveur créé en 2009
- Serveur disponible sous Windows MAC Linux ...
- Node.js peut faire tourner des applications diverses : back, desktop, web...
- Basé sur le moteur Javascript V8
- 2 versions : une LATEST et une **LTS**
- Utilisée par GoDaddy, IBM, NetFlix, Amazon Web Services, Groupon, Vivaldi, SAP, LinkedIn, Microsoft, Yahoo!, Walmart, Rakuten, Sage et PayPal1.

Node.js
Généralités

- Navigateur
  - JS dépend du navigateur
  - Accès aux objets document et windows
  - Manipulation du DOM
  - **import** pour charger un module
- Node.js
  - Gestion de l'environnement
  - JS dépends de la version de serveur
  - **require** pour charger un module
  - Accès au système de fichier

1ere application

test/HelloWorld.js

```
console.log(“HelloWorld!”);
```

Exécuter le code : Dans la console (powershell, shell ou bash) lancez la commande suivante : node HelloWorld.js

Quitter le programme :

- OS : ctrl+C dans la console
- En js :
  - process.exit() : arrêt brutal et perte des tâches asynchrones
  - Utilisation de SIGTERM

calc/calc.js

```
var sum = function (a, b) {
  return a + b;
};
console.log('ligne troll');
module.exports = {
  sum: sum
}; //ou module.exports.sum=sum;
```

calc/index.js

```
var calc = require('./calc.js');

var result = calc.sum(1, 3);
console.log(result);
```

Module :

- En JS (et donc Node), chaque fichier est un module.
- module.exports une variable ou un objet
- require(*'moduleName'*) pour importer le module
- l'export exécute le code du fichier JS ciblé.

NPM part1

console cd /…/calc
npm init
npm install express --save

calc/index.js
var express = require('express');

NPM: node package manager

- npm init : génère un package.json pour votre projet
- npm install *PackageName* : installe le package
- npm install PackageName --save : installe le package et l'ajoute à package.json de votre projet
- npm install -g PackageName : pour installer GLOBALEMENT un package
- require('PackageName'); pour importer le module du package

Express

console cd /.../calc
npm init
npm install express --save

calc/index.js
var express = require('express');

Express: serveur http simplifié

- Node possède un module *http* préinstallé mais lourd à utiliser
- Express est plus facile et nécessite moins de code

test/index.js

```
var express = require('express');
var app = express();

app.get('/', function (req, res) {
  res.send('Hello World!');
});

//ici l'app est créée mais n'est pas "exposé"

app.listen(3000, function () {
  console.log('Example app listening on port 3000!');
});
```

Express: serveur http simplifié

- app=express : nouvelle appli web/http
- .get : on répond à la requête http GET
- / : la route(url) à laquelle on va répondre
- req : objet correspondant à la requête
- res : la réponse de notre serveur à cette requête
    - res.send() : envoi une réponse varié
    - res.json() : envoi une réponse en JSON
    - res.render() : envoi un template
    -

calc/index.js

```
var express = require('express');
var app = express();

app.get('/', function (req, res) {
  res.send('API calculatrice');
});

app.get('/add/:input1/:input2', function (req, res) {
  var input1 = parseInt(req.params.input1);
  var input2 = parseInt(req.params.input2);

  var result = input1 + input2;

  res.send(result.toString());
});

app.listen(3000, function () {
  console.log('Example app listening on port 3000!');
});
```

Gérer plusieurs route

- 2 ".get" qui répondent à 2 routes différentes
- :input1 : paramètre de route dans l'url directement
- parseInt : traduit un String en Int
- req.params.input1 :
  paramètre de route **input1** dans l'objet **params** de la variable **req**

calc/index.js

```
var express = require('express');
var app = express();

var calc = require('./calc.js');

app.get('/', function (req, res) {
  res.send('Hello World!');
});

app.get('/add/:input1/:input2', function (req, res) {
  var input1 = parseInt(req.params.input1);
  var input2 = parseInt(req.params.input2);

  var result = calc.sum(input1, input2);

  res.send(result.toString());
});

app.listen(3000, function () {
  console.log('Example app listening on port 3000!');
});
```

Gérer plusieurs route

- 2 ".get" qui répondent à 2 routes différentes
- :input1 : paramètre de route dans l'url directement
- parseInt : traduit un String en Int
- req.params.input1 :
  paramètre de route **input1** dans l'objet **params** de la variable **req**

TP

Construisez 3 autres modules pour la soustraction la multiplication et la division

Ajoutez 3 nouvelles routes qui appelleront ces modules