

# Panorama des langages scripts

## Introduction à Node.js

Tom MAFILLE

### Premier serveur : la suite

1. Servir une page html

```
var PORT = 8080;
var http = require('http');

var serv = http.createServer(function (req, res) {
  res.writeHead(200, { 'Content-Type': 'text/html' });
  res.write(
    '<!DOCTYPE html>' +
    '<html lang="fr">' +
    '<head>' +
    '<title>Mon premier serveur node.js</title>' +
    '<meta charset="utf-8" />' +
    '</head>' +
    '<body>' +
    '<h1>Hello</h1>' +
    '<p>Ceci est ma première page html.</p>' +
    '</body>' +
    '</html>'
  );
  res.end();
});

serv.listen(PORT);
console.log("Server running on " + PORT);
};
```

2. Installer un module non standard

```
npm install swig
```

- Fichier js 1ere version :

```
var PORT = 8080;
var swig = require('swig');
var http = require('http');

var serv = http.createServer(function (req, res) {
  res.writeHead(200, { 'Content-Type': 'text/html' });
  res.write(swig.renderFile('templates/home.tpl', {
    name : 'user'
  }));
  res.end();
});
```

```
serv.listen(PORT);
console.log("Server running on " + PORT);
```

- Fichier .tpl :

```
<doctype html>
<html lang="fr">
  <head>
    <title>Mon premier serveur node.js</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1>Hello {{ name }} !</h1>
    <p>Ceci est ma première page html</p>
  </body>
</html>
```

- Fichier server\_module.js :

```
var http = require('http');
var swig = require('swig');
exports.startServer = function (port) {
  var server = http.createServer(function(req, res) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(swig.renderFile('templates/home.tpl', {
      name : 'user'
    }));
  });
  server.listen(port);
  console.log('server running on ' + port);
}
```

- Fichier js deuxième version :

```
var PORT = 8080;
var server = require('./server_module');
server.startServer(PORT);
```

3. Servir des pages différentes en fonction de l'URL : Dans server\_module.js

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

exports.startServer = function (port) {
  var server = http.createServer(function(req, res){ //definition de la fonction
    var page = url.parse(req.url).pathname; // Récupère le chemin de l'URL

    if (page === '/') { // Si l'URL est "/"
      res.writeHead(200, {'Content-Type': 'text/html'}); //ecrire en html
      res.write(swig.renderFile('templates/home.tpl', { //Contenu du template
        name: 'user' //le nom
      }));
    } else {
      if (page === '/toto') { // Si l'URL est "/toto"
        res.writeHead(200, {'Content-Type': 'text/html'}); //Ecrire en html
        res.write(swig.renderFile('templates/home.tpl', { //Le contenu du template
          name: 'TOTO LE RIGOLO' //le NOM
        }));
      }
    }
  });
  server.listen(port);
  console.log('server running on ' + port);
}
```

```

    }));
  } else {
    // Pour toutes les autres pages
    res.writeHead(404, {'Content-type': 'text/html'}); //écrire en html
    res.write('<h1> Error 404 : page not found </h1>'); //erreur 404
  }
}
res.end(); // Termine la réponse
});

server.listen(port);
console.log('server running on ' + port);
};

```

#### 4. Récupérer des données transmises en GET

```

var http = require('http');
var swig = require('swig');
var url = require('url');
var querystring = require('querystring');

exports.startServer = function (port) {
  var server = http.createServer(function(req, res) {
    var page = url.parse(req.url).pathname; // Récupère le chemin de l'URL

    if (page === '/') {
      var params = querystring.parse(url.parse(req.url).query);
      var data = { name: 'unknown user' }; // Valeur par défaut pour 'name'
      if ('name' in params) { // Si le paramètre 'name' existe
        data['name'] = params['name']; //on le prend en paramètre
      }
      res.writeHead(200, { 'Content-Type': 'text/html' });
      res.write(swig.renderFile('templates/home.tpl', data)); //ecris le template
      res.end(); // Termine la réponse
    } else {
      res.writeHead(404, { 'Content-Type': 'text/html' });
      res.write('<h1>Error 404: Page not found</h1>');
      res.end();
    }
  });

  server.listen(port);
  console.log('Server running on port ' + port);
};

```

#### 5. Récupérer des données transmises en POST

- Modifications dans home.tpl :

```

<!DOCTYPE html>
<html lang="fr">
<head>
  <title>Mon premier serveur node.js</title>
  <meta charset="utf-8" />
</head>
<body>
  <h1>Hello {{ name }} !</h1>
  <p>Ceci est ma première page html</p>
  <form action="message" method="POST">

```

```

    <fieldset>
      <legend>Message pour le serveur</legend>
      <textarea rows="10" cols="40" name="msg"></textarea>
      <input type="submit" value="Envoyer!" />
    </fieldset>
  </form>
</body>
</html>

```

- La page chargée qui permet de gérer le formulaire est /message
- Données transmises par la méthode POST
- Fichier modifié :

```

var http = require('http');
var swig = require('swig');
var url = require('url');
var querystring = require('querystring');

function error404(res) {
  res.writeHead(404, { 'Content-Type': 'text/html' });
  res.write('<h1>Error 404 : page not found </h1>');
  res.end();
}

exports.startServer = function (port) {
  var server = http.createServer(function (req, res) {
    var page = url.parse(req.url).pathname;

    if (page === '/') {
      var params = querystring.parse(url.parse(req.url).query);
      var data = { name: 'unknown user' };
      if ('name' in params) {
        data['name'] = params['name'];
      }
      res.writeHead(200, { 'Content-Type': 'text/html' });
      res.write(swig.renderFile('templates/home.tpl', data));
      res.end();
    } else if (page === '/message') {
      if (req.method === 'POST') {
        var post_data = '';
        req.on('data', function (p_data) {
          post_data += p_data;
        });
        req.on('end', function () {
          var final_data = querystring.parse(post_data);
          if (final_data['msg'] !== "") {
            console.log('RECU: ' + final_data['msg']);
            res.writeHead(200, { 'Content-Type': 'text/html' });
            res.write('<h1>POST</h1><p>Données bien reçues</p>');
            res.end();
          } else {
            console.log('Absence de données!');
            res.writeHead(200, { 'Content-Type': 'text/html' });
            res.write('<h1>POST</h1><p>Aucune donnée</p>');
            res.end();
          }
        });
      }
    }
  });
}

```

```

        }
    });
    } else {
        error404(res);
    }
} else {
    error404(res);
}
});

// Démarrage du serveur
server.listen(port, function () {
    console.log('Server running on port ' + port);
});
};

```

## BONUS :

### 1. App.js :

```

const express = require('express');
const bodyParser = require('body-parser'); // Middleware pour gérer les formulaires
const app = express();

// Middleware pour les formulaires
app.use(bodyParser.urlencoded({ extended: true }));

// Définir le moteur de rendu EJS (ou tout autre moteur)
app.set('view engine', 'ejs');
app.set('views', './views');

// Route principale
app.get('/', (req, res) => {
    res.send('Bienvenue sur la page principale !');
});

// Nouvelle route /user
app.get('/user', (req, res) => {
    res.render('user_form');
});

// Gestion de la soumission du formulaire
app.post('/user', (req, res) => {
    const name = req.body.name;
    res.send(`<h1>Bonjour, ${name} !</h1>`);
});

// Démarrer le serveur
const port = 3000;
app.listen(port, () => {
    console.log(`Serveur démarré sur http://localhost:${port}`);
});

• user_form.ejs:
<!DOCTYPE html>
<html lang="fr">
<head>

```

```
<title>Formulaire Utilisateur</title>
<meta charset="UTF-8">
</head>
<body>
  <h1>Entrez votre nom</h1>
  <form action="/user" method="POST">
    <label for="name">Nom :</label>
    <input type="text" id="name" name="name" required>
    <button type="submit">Envoyer</button>
  </form>
</body>
</html>
```