Utiliser des templates : l'exemple d'EJS

- Utiliser Express tel quel pour la génération des pages HTML pose de gros problèmes, en mélangeant code JavaScript et HTML.
- Il est tout à fait possible, comme en Java ou en PHP, de recourir à des systèmes de templates.
- Express est compatible avec la plupart des moteurs de template comme Smarty, Twig, etc.
- L'utilisation du système EJS est argumentée parce qu'il repose sur la syntaxe de JavaScript.
- L'installation se fait en tapant :

npm install ejs --save

Mise en œuvre de base

- Tout d'abord, il faut indiquer à node que le moteur de rendu ne sera pas celui par défaut, mais qu'il faudra au préalable passer par une phase d'interprétation avec EJS :

```
var app = express() ;
app.set ('view-engine', 'ejs') ;
```

- Il faut créer, à la racine de l'application, un répertoire s'appelant obligatoirement **views**.
- Ensuite, pour chacune des pages vers lesquelles le routeur pourrait être amené à rediriger et pour lesquelles on souhaite recourir au système de template, il faut créer le gabarit de page correspondant dans le répertoire en question, et indiquer que la réponse doit être « rendue » avec le gabarit de page correspondant :

```
app.get('/page1', function(req, res){
  res.render('page.ejs') ;
}) ;
```

- Avec EJS (comme les autres modèles express), on peut exécuter du code serveur et accéder aux variables serveur à partir du code HTML.

- Dans EJS, on utilise " **<%** " comme balise de début et " **%>** " comme balise de fin, les variables transmises comme les paramètres de rendu sont accessibles avec **<%=var_name%>**
- ***Exemple 1*** *:*

Vous êtes au niveau numéro **<%= niveau %>**

- ***Exemple 2 :*** Passage de paramètre depuis l'URL

De la même manière qu'on peut passer des paramètres directement à l'aide d'Express, on peut en passer à EJS :

```
var express = require('express');
var app = express();
app.set ('view-engine', 'ejs') ;
app.get('/page:num', function(req, res){
  res.render('page.ejs', {"numero": req.params.num}) ;
}) ;
var server = app.listen(8081, function () {
     var port = server.address().port
     console.log("Serveur sur le port  "+port) })
```

Dans le gabarit views/ page.ejs, on écrira :

```
<!DOCTYPE html >
<html lang="fr">
<head>    <title>Page de test</title>      <meta charset="utf-8" /></head>
<body>
            <p>Ceci est un paragraphe, sur la page <%= numero %>.</p>
</body>
</html>
```

<%= numero %> est remplacé par la valeur de la variable numero qui a été transmise au gabarit via req.params.num

Condition if

- Dans cet exemple, dans page.ejs, la condition déclare si la variable d'âge est supérieure à 20, le message « valide » s'affiche, autrement « non autorisé » sera affiché.

```
<% if (age > 12) { %>
<p> valide !</p>
<%} else { %>
<p> non autorisé </p>
<% } %>
```

- Lors de l’appel du template :

```
app.get("/test", function(req, res) {
res.render("page", { age: 16 });
});
```

Boucle for

- si on a une baie de consommables dans le code serveur, par exemple, on peut le parcourir en utilisant un boucle,
- On veut créer li à l'intérieur de la boucle for. On doit fermer l’étiquette EJS à la fin du for et créer une nouvelle balise juste pour les accolades,

```
<h1><%= title %></h1>
  <ul>
<% for(var i=0; i<tab.length; i++) { %>
   <li>
      <a href='page/<%= tab[i] %>'>
         <%= tab[i] %>
      </a>
   </li>
<% } %>
```

- ***Exemple :***

```
var express = require('express');
var app = express();
app.set ('view-engine', 'ejs') ;
app.get('/page:num', function(req, res){
  res.render('page.ejs', {"numero": req.params.num}) ; }) ;
 var server = app.listen(8081, function () {
     var port = server.address().port
     console.log("Serveur sur le port  "+port) })
```

Page.ejs :

```
<!DOCTYPE html >
<html lang="fr">
<head>   <title>Page de test</title>     <meta charset="utf-8" /> </head>
<body>
            <p>Ceci est un paragraphe, sur la page  numéro  <%= numero %>.</p>
            <ul>
                        <% for (let i=0;i<numero;i++){%>
                                    <li>Ligne <%= i+1%></li>
                        <%}%>
            </ul>
</body></html>
```

Boucle forEach

```
<ul>
    <% boissons.forEach(function(boisson) { %>
        <li><%= boisson.nom %> - <%= boisson.calories %></li>
    <% }); %>
</ul>
```

Include

```
<%- include('header'); -%>
<h1>
  Title
</h1>
<p>
  My page
</p>
<%- include('footer'); -%>
```

Routage avec Node.js express et ejs

- Le but de cet exemple est de créer un site Web de 3 pages avec des pages d'accueil, à propos et de contact avec des modèles d'en-tête, de navigation et de pied de page et le routage du trafic par le serveur express.
- On crée d'abord un dossier de projet principal et on exécute npm init. Ensuite, on l'organise comme ci-dessous.

-- project_folder
 -- css
  -- style.css
 -- html
  -- about.ejs
  -- contact.ejs
  -- index.ejs
  -- include
   -- footer.ejs
   -- head.ejs
   -- nav.ejs
   -- scripts.ejs
 -- routes
  -- index.js
 -- app.js

Packages

Pour le front-end, on utilisera :
bootstrap 4 qui nécessite jquery et popper.js.

Le dossier principal du projet

Pour servir des fichiers statiques, on doit utiliser express.static. Par défaut, express ne peut pas accéder aux fichiers des autres dossiers. On doit également définir le répertoire des vues par défaut dans lequel on a les fichiers ejs. Le routeur est importé et utilisé comme middleware.

Fichier app.js

```
const express = require('express');
const path = require('path');
const routes = require('./routes');
const app = express();
app.set('views', path.join(__dirname, 'html'));
app.use(express.static(path.join(__dirname,'css')));
app.use(express.static(path.join(__dirname, 'node_modules')));
app.set('view engine', 'ejs');

app.use('/', routes);

app.listen(3000, () => {
  console.log('Server is running at localhost:3000');
});
```

Routage

On utilisera la classe Router. Cela nous permet d'exporter la logique de routage en tant que module et de l'utiliser dans le principal app.js. Lorsque la logique se complique, l'utilisation de la classe Router pour compartimenter le routage facilite la maintenance.

Fichier index.js

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

router.get('/', (req, res) => {
  console.log('Request for home recieved');
  res.render('index');
});

router.get('/about', (req, res) => {
  console.log('Request for about page recieved');
  res.render('about');
});

router.get('/contact', (req, res) => {
  console.log('Request for contact page recieved');
  res.render('contact');
});
module.exports = router;
```

Les templates ejs

Fichier index.ejs

```
<!DOCTYPE html>
<html lang="en">
    <%- include ('./include/head') %>
    <body>
        <%- include ('./include/nav') %>
        <div class="container">
            <div class="row">
                <h1>Home</h1>
            </div>
            <hr>
            <div class="row">
                <p style="height:300px;">Contenu de la page ici...</p>
            </div>
        </div>

        <%- include ('./include/scripts') %>
        <%- include ('./include/footer') %>
    </body>
</html>
```

Fichier about.ejs

```
<!DOCTYPE html>
<html lang="en">

    <%- include ('./include/head') %>
    <body>
        <%- include ('./include/nav') %>
        <div class="container">
            <div class="row">
                <h1>About</h1>
            </div>
            <hr>
            <div class="row">
                <p style="height:300px;">Contenu de la page ici...</p>
            </div>
        </div>

        <%- include ('./include/scripts') %>
        <%- include ('./include/footer') %>
    </body>
</html>
```

Fichier contact.ejs

```
<!DOCTYPE html>
<html lang="en">
    <%- include ('./include/head') %>
    <body>
        <%- include ('./include/nav') %>
        <div class="container">
            <div class="row">
                <h1>Contact</h1>
            </div>
            <hr>
            <div class="row">
                <p style="height:300px;">Contenu de la page ici...</p>
            </div>
        </div>

      <%- include ('./include/scripts') %>
        <%- include ('./include/footer') %>
    </body>
</html>
```

Fichier head.ejs

```
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <title>Node EJS | Home</title>

    <link rel="stylesheet" type="text/css"
href="../node_modules/bootstrap/dist/css/bootstrap.min.css">
   <link rel="stylesheet" type="text/css" href="style.css">
</head>
```

Fichier nav.ejs

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse"
    data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false"
aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
        <div class="navbar-nav ml-auto">
        <a class="nav-item nav-link active" href="/">Home</span></a>
        <a class="nav-item nav-link" href="about">About</a>
        <a class="nav-item nav-link" href="contact">Contact</a>
        </div>
    </div>
</nav>
```

Fichier footer.ejs

```
<footer class="container-fluid footer">
  <div class="container">
    <div class="row">
      <div class="col-12">
        <p></p>
        <p></p>
        <% let year = (new Date()).getFullYear() %>
        <p style="float:right;">&copy;<%= year %> </p>
      </div>
    </div>

  </div>
</footer>
```

Fichier script.ejs

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-
J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
crossorigin="anonymous"></script>
```

Fichier style.css

```
hr {
  background-color: #fff;
        border-top: 2px dotted #8c8b8b;
}

.footer{
  background-color: #D1C4E9;
  margin:0px auto;
  padding: 20px 0px 20px 0px;
  vertical-align:bottom;
}
```