Templating

NodeJS

Templating

NodeJS seul n'est pas très pratique pour afficher de belle pages Html car nous devons encoder tout notre html dans un string.

```
let body = `<!doctype>
<html><body>
    <form action="/" method="post">
        <input type="text" name="fname" /><br />
        <input type="number" name="age" /><br />
        <input type="file" name="photo" /><br />
        <button>Save</button>
    </form>
</body></html>`;
```

Nous avons bien entendu la possibilité d'utiliser une Framework comme express (https://expressjs.com/fr/) qui nous faciliterais la vie pour le templating mais également pour la gestion du serveur web.

Mais si nous désirons simplement utiliser un moteur de Template sans Framework, c'est possible !!!

Il existe plusieurs moteurs de template : Vash, Jade, Haml, Mustache, Handlebar, Underscore,Pure,…

Pour notre exemple, nous utiliserons EJS (https://ejs.co/)

Templating

EJS

Pourquoi EJS ?

Car c'est un langage de template simple qui permet de générer un balisage Html via le javascript.

Quelques infos de syntaxe :

Tag	Utilisation
<%	Juste pour inclure du script, aucune sortie html
<%_	Supprime tous les espaces avant la valeur à afficher
<% =	Affiche la valeur du modèle envoyé (Html escaped)
<% -	Affiche la valeur Html non escaped
<%%	Affiche un littéral
<% #	Commentaire
%>	Balise de fin

EJS

Include

Il est possible d'inclure un morceau de template dans un autre (~= partial view).

Pour cela, nous utiliserons la syntaxe suivante :

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

Le paramètre de la fonction *include* est une chemin relatif vers le fichier ejs à inclure

Templating

Ejs – Utilisation

Première étape, installer le package via npm

```
npm install ejs
```

Ensuite, nous créons notre template aevc l'extension *ejs*

```
<!doctype>
<html>
    <body>
        <h1><%= titreform %></h1>
        <form action="/" method="post">
            <input type="text" name="fname" /><br />
            <input type="number" name="age" /><br />
            <input type="file" name="photo" /><br />
            <button>Save</button>
        </form>
    </body>
</html>
```

Templating

Nous pouvons finalement utiliser ce template dans notre script,

Pour cela, nous devons charger la librairie :

```
const ejs = require('ejs');
```

Grâce à cette librairie, nous avons accès à la fonction *render(template, Model).*

- Template : un string qui contient l'html avec les séquence <% permettant l'injection des valeurs du Model
- Model : un objet avec les différents clé/valeurs devant être injectées dans le template

Pour créer le string qui sera associé au template, nous utiliserons *fs.readFileSync* pour récupérer le contenu de notre View

Templating

Remarque :

si *ejs.render* ne se trouve pas dans le fichier d'entrée principale (app.js par exemple), les includes ne fonctionnerons pas car ejs ne pourra pas retrouver le chemin relatif au document.

Il faut alors utiliser

```
ejs.renderFile(filename, data, options, function(err, str){ // str => Rendered HTML string });
```

(https://github.com/mde/ejs)