

TD/TP JavaScript 2 : Invocation d'un serveur Node.js

Pierre Pompidor

Créez un nouveau dossier pour ce second TP JavaScript (pour ne pas mélanger les codes des deux TP).

Depuis le Moodle, téléchargez le contenu du dossier TD/TP "AJAX : invocation d'un serveur Node.js" :

- Le code HTML, la feuille de style CSS et les différentes photos de James Bond
- le fichier contenant la liste d'objets JavaScript (*cocktails.json*)
- Le serveur Node.js (*serveur_v1.js*)

Pour dézipper dans votre terminal l'archive que vous avez obtenue : `unzip <nomDeLArchive>`

Copiez dans votre dossier le code JavaScript que vous avez développé pour le premier TP sous le nom *CV_James_AJAX.js*.

Test du serveur Node.js :

Lancez le serveur Node.js depuis votre terminal : `node serveur1.js`

Depuis votre navigateur, invoquez le serveur sur les deux URLs suivantes :

- `localhost:8888` : le CV "déformaté" et sans photo devrait apparaître;
- `localhost:8888/cocktails` : les informations concernant les cocktails devraient apparaitre.

Analyser le code HTML et les erreurs qui s'affichent dans la console pour expliquer le fait que le CV soit si bizarre.

Rajout du service web qui renvoie la feuille de style et les images :

Arrêtez le serveur (Control-C) et rajoutez-y le service web qui permet de renvoyer un fichier correspondant soit à la feuille de styles, soit à une image, puis redémarrez le...

La route qui invoque ce service web doit donc comporter un paramètre dont la valeur sera le nom du fichier que le serveur doit renvoyer...

Modification de la fonction `afficherCocktails()` :

Modifiez la fonction *afficherCocktails()* pour obtenir les informations concernant les cocktails depuis serveur Node.js.

Filtre des cocktails sur le serveur et non sur le client :

Nous voulons maintenant que le serveur ne renvoie que les noms des cocktails bus par le James Bond "courant" (celui dont la photo est affichée).

Cela implique :

- de modifier la fonction *afficherCocktails()* pour que celle-ci invoque le serveur via une route comportant le nom du James Bond, par exemple : `/cocktails/Connery`
- de modifier le service web qui renvoie les informations sur les cocktails pour que celui-ci :
 - soit invoqué sur une route comportant le nom de James
 - déserialise le JSON du fichier *cocktails.json* avec *JSON.parse()*
 - construise une liste avec les noms des cocktails
 - et serialise la liste avant de la renvoyer avec *JSON.stringify()*
- la fonction *afficherCocktails()* doit donc aussi gérer une liste de noms de cocktails...