

Programmation en JavaScript - TP n.1

-

Intégration dynamique de données dans le DOM / AJAX

—

Pierre Pompidor

Préambule

Ce premier TP va traiter de l'intégration dynamique de données en mémoire du navigateur (dans le DOM), et en conséquence la modification du rendu de la page HTML. Il nécessite l'utilisation d'un serveur JavaScript exécuté par **Node.js** qui enverra au navigateur les données à intégrer. Comme vu en cours, Node.js (de petit nom *Node*) est un interpréteur JavaScript orienté réseau généralement exécuté sur un serveur.

Installation de Node.js

Sous Linux/Ubuntu

```
sudo apt-get update  
sudo apt-get install nodejs npm
```

Sous Windows

Téléchargez Node.js <https://nodejs.org/fr/> et installez-le

Test de Node.js

Que ce soit sous Linux ou Windows, créez un fichier nommé *bonjour.js* :

```
console.log("Bonjour !")
```

Dans un terminal Linux ou dans une invite de commandes DOS, vérifiez que vous pouvez bien exécuter ce programme avec Node :

```
node bonjour.js
```

Installation du module express

Pour que notre serveur Node puisse renvoyer des données à partir de routes (des messages spécifiques émis par le navigateur à destination du serveur Node), installez le module *express* (par la même commande sous Linux ou Windows) :

```
npm install express
```

Utilisation de la bibliothèque JavaScript JQuery

Nous utiliserons principalement la bibliothèque JavaScript **JQuery** pour invoquer le serveur Node et allouer en mémoire du navigateur les données qu'il renvoie (objets JavaScript). L'utilisation de cette bibliothèque n'est pas une obligation mais elle permet de programmer de manière plus concise. Il est inutile de la télécharger (elle sera importée dynamiquement).

Enrichissement dynamique du CV de James

Retour sur l'exemple vu en cours

Reprenez les codes concernant le CV de James vus en cours (disponibles sur le moodle dans les archives *CV_James.tar.gz* ou *CV_James.zip*).

Exécutez le serveur Node fourni (*serveur_express.js*) qui renvoie une liste d'objets formatée en JSON (*JavaScript Object Notation*);.

Invoquez le serveur Node à partir de votre navigateur - n'oubliez pas qu'il s'exécute par défaut sur le port 8888 - et vérifiez que celui-ci affiche les données renvoyées par le serveur.

Stupéfiantes améliorations

Mise en place d'un carrousel d'images

Téléchargez et dézippez à partir du moodle le dossier contenant les différentes images de James (*Images_James.tar.gz* ou *Images_James.zip*).

Modifiez les codes HTML et JavaScript de telle sorte que lors d'un clic souris sur l'image de James, cela soit la prochaine qui soit affichée (cet affichage devant être circulaire). Pour ce faire, vous devez créer une liste contenant les noms des différentes images.

Intégration sélective des cocktails préférés de James

Modifiez le code JavaScript de telle sorte que la sélection du lien sur les cocktails préférés n'affiche que ceux appréciés par l'avatar courant de James. Attention : si une liste d'alcools était déjà affichée, celle-ci doit être remplacée par la nouvelle.

Quelques indications techniques :

- La méthode JavaScript *split()* permet de découper une chaîne de caractères par rapport à un délimiteur ;
- La méthode JavaScript *includes()* permet de vérifier la présence d'une valeur dans une liste ;
- La méthode JQuery *empty()* permet de vider le contenu d'un élément du DOM.

Création d'un catalogue musical

La seconde partie du TP concerne l'affichage "récuratif" d'un catalogue musical :

- les données correspondant aux chanteurs seront chargées automatiquement lors du chargement de la page HTML ;
- lors de la sélection de l'image d'un chanteur, les données correspondant à ses albums seront chargées ;
- lors de la sélection de l'image d'un album, les données correspondant aux titres de ses albums seront chargées (et nous pourrions imaginer que des liens sur des mp3 soient proposés).

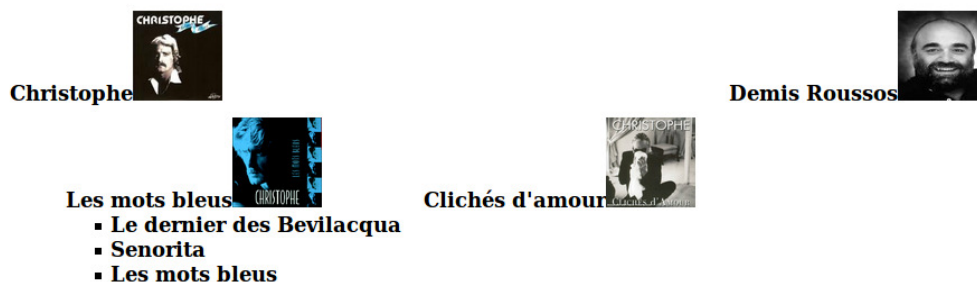
La page HTML est chargée, les images des chanteurs apparaissent :



l'internaute clique sur l'image de Christophe, ses magnifiques albums apparaissent :



l'internaute clique sur la pochette de l'album "les mots bleus", les envoûtants titres de celui-ci apparaissent :



Les différentes données seront renvoyées par le serveur Node *serveur_donnees_musicales.js* :

`http://localhost:8888/chanteurs` → données correspondant aux chanteurs

`http://localhost:8888/albums/<nomDuChanteur>` → données correspondant aux albums du chanteur

`http://localhost:8888/album/<nomDeLAlbum>` → données correspondant à un album

Vous remarquerez que ci-dessus les chaînes de caractères entourées par des chevrons correspondent à des paramètres (par exemple `<nomDuChanteur>` pourrait être *Christophe* ou *Roussos*).

Exemples de fichiers au format JSON

chanteurs.json :

```
[
  {
    "nom": "Christophe",
    "prenom_reel": "Daniel",
    "nom_reel": "Bevilacqua",
    "image": "christophe.jpg",
    "lien": "albums-Christophe"
  },
  {
    "nom": "Demis Roussos",
    "prenom_reel": "Artemios",
    "nom_reel": "Roussos",
    "image": "roussos.jpg",
    "lien": "albums-Roussos"
  }
]
```

albums-Christophe.json :

```
[
  {
    "nom": "Les mots bleus",
    "image": "Les_mots_bleus.jpg",
    "lien": "album-les_mots_bleus"
  },
  {
    "nom": "Clichés d'amour",
    "image": "Cliches_d_amour.jpg",
    "lien": "album-cliches_d_amour"
  }
]
```

Etapes de réalisations

Téléchargez et dézippez à partir du moodle le dossier contenant le serveur Node, la page HTML, les fichiers JSON et les images (*Catalogue_musical.tar.gz* ou *Catalogue_musical.zip*).

Voici la page HTML initiale (*catalogue_musical.html*) :

(Vous remarquerez que le body est vide.)

```
<html>
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <script src="https://code.jquery.com/jquery-3.5.1.min.js" integrity="sha256-9/aliU8dGd2tb60Ssu" />
    <script src="catalogue_musical.js"></script>
  </head>
  <body/>
</html>
```

Exécution du serveur Node

Lancez le serveur Node qui va envoyer les données JavaScript sérialisées :
(a priori ce serveur ne sera pas à modifier)

```
node serveur_donnees_musicales.js
```

Testez ce serveur en l'invoquant suivant différentes routes depuis votre navigateur.

Affichage des informations liées aux chanteurs

Créez dans un fichier *catalogue_musical.js* le code JavaScript qui permet d'invoquer le serveur pour récupérer les données correspondant aux chanteurs et les intégrer au DOM du navigateur.

Ce code doit mettre en œuvre la méthode *getJSON()* de JQuery.

```
$(document).ready(function() { // Le code de la fonction de callback
    // ne sera exécuté que quand les éléments du DOM auront été alloués
    ...
    $.getJSON(..., function(data) {
        ...
    });
});
```

Par rapport aux données du fichier *chanteurs.json*, le code JavaScript doit avoir le même effet qui si nous avions directement codé les instructions HTML suivantes.

Attention : le code ci-après n'est donc pas à saisir dans la page HTML !

```
<ul id="listePrincipale">
  <li> Christophe 
    <ul id="liste_albums-Christophe" />
  </li>
  <li> Demis Roussos 
    <ul id="liste_albums-Roussos" />
  </li>
</ul>
```

Affichage des informations liées aux albums d'un chanteur ou à un album particulier

Lors de la sélection d'une image (avec un clic), appelez une fonction *chargement()* qui va solliciter le serveur pour récupérer les données et les intégrer dans le DOM du navigateur pour mettre à jour la page.

Cette fonction doit aussi bien gérer les données liées aux albums d'un chanteur que celles d'un album particulier.

Cette fonction doit aussi mettre en œuvre la méthode *getJSON()* de JQuery.

```
function chargement() {
    ...
    $.getJSON("http://localhost:8888/"+..., function(data) {
        ...
    });
}
```

Pour aller plus loin

Il serait habile et gracieux qu'un nouveau clic sur une image "referme" les informations affichées...

Cela passerait par :

la sélection d'éléments du DOM avec la fonction *\$(<selecteur CSS>)* de JQuery

ainsi que les méthodes *hide()* et *show()*.