

Web 2.0

TP2 : Requêtes dynamiques

1. Introduction

L'objet de ce TP est d'intégrer les requêtes développées en TP1 (recherche d'auteurs par nom, recherche d'ouvrages par titre, etc.) dans un programme PHP. Le but est d'exécuter ce programme à travers une fonction JavaScript (AJAX) afin d'afficher les résultats dynamiquement et sans quitter la page web courante.

2. Recherche d'auteurs par nom

Rappel : dans le TP1, la balise `<section>` centrale (où est affiché "Bienvenue sur le site ...") a été partagée en une `<div>` gauche et une `<div>` droite (selon le fichier CSS).

1) Créer et tester un programme *recherche_auteurs.php* qui recherche les auteurs dont le nom contient le paramètre *debnom*. Ce fichier renvoie un document JSON structuré de la façon suivante :

```
[{"code":266,"nom":"Ambrose","prenom":"David"},
 {"code":182,"nom":"Anderson","prenom":"Kevin J."},
 ...
]
```

Utiliser :

- `header("Content-Type: application/json; charset=UTF-8") ;`
- [PDO](#) (PHP Data Objects) pour accéder à la base de données depuis PHP (recommandé).
- `query()` (pour exécuter la requête SQL retournant les auteurs à partir de la BDD)
- `fetchAll()` (pour récupérer les données des auteurs à partir de la BDD)
- `json_encode()` (pour encoder les données en format JSON affichées avec `echo`) : [JSON PHP](#)

2) Sur événement *onkeyup* de la 1ère balise `<input>` du menu, appeler une fonction AJAX *recherche_auteurs()* qui lance l'exécution du programme *recherche_auteurs.php* paramétré par *debnom* dont la valeur provient de la `<input>`. Les données au format JSON retournées par *recherche_auteurs.php* seront "parsées" dans la fonction callback *affiche_auteurs(...)* pour obtenir un tableau (array) d'auteurs. Celui-ci permettra de créer une liste ordonnée d'auteurs comme suit :

```
<ol>
  <li><a href="#" onclick="recherche_ouvrages_auteur(code)">Nom Prénom</a></li>
  ...
</ol>
```

Cette liste devra être attachée à la `<div>` gauche (retrouvée à partir de son *id*).

- Dans un 1^{er} temps, afficher uniquement le *Nom Prénom* de chaque auteur (sans les liens).
- La fonction AJAX *recherche_ouvrages_auteur(code)* devra être définie en paragraphe « 4. Recherche d'ouvrages par auteur ».

Utiliser :

- JavaScript : `getElementById`, `innerHTML` et le (+) pour générer les listes de liens ; [JSON HTML](#)
- HTML : attribut événements `onclick`, `onkeyup`,
- AJAX : `XMLHttpRequest`, `responseText`, `onreadystatechange`, `open()`, `send()` ... ; [AJAX](#)
- JSON : `JSON.parse(req.responseText)` où *req* est l'objet de type `XMLHttpRequest` ; [JSON Parse](#)

Web 2.0

TP2 : Requêtes dynamiques

3. Recherche d'ouvrages par titre

1) Créer et tester un programme *recherche_ouvrages_titre.php* qui recherche les ouvrages (et les exemplaires associés) dont le titre contient le paramètre *debtitre*. Ce fichier renvoie un document JSON structuré de la façon suivante :

```
[{"code":392,"nom":"Dangereuses visions I",
  "exemplaires":[{"nom":"J'ai lu","code":404,"prix":"0.50"}]
},
{"code":397,"nom":"Dangereuses visions II",
  "exemplaires":[{"nom":"J'ai lu","code":409,"prix":"0.50"}]
},
...
]
```

Utiliser : les mêmes références que celles proposées dans la question 2. **Recherche par auteur**

2) Sur événement *onkeyup* de la 2ème balise *<input>* du menu, appeler une fonction AJAX *recherche_ouvrages_titre()* qui lance le programme *recherche_ouvrages_titre.php* paramétré par *debtitre* dont la valeur provient de la *<input>*. Les données au format JSON retournées par *recherche_ouvrages_titre.php* seront "parsées" dans la fonction *callback affiche_ouvrages(...)* pour obtenir un tableau (array) d'ouvrages. Celui-ci permettra de créer une liste ordonnée d'ouvrages et, pour chacun, une liste d'exemplaires comme suit :

```
<ol>
  <li> titre de l'ouvrage
    <ul>
      <li>Nom editeur1, prix1 euros</li>
      <li>Nom editeur2, prix2 euros</li>
      ...
    </ul>
  </li>
  ...
</ol>
```

Cette liste devra être attachée à la *<div>* située à droite (retrouvée à partir de son *id*).

Dans un 1^{er} temps, afficher uniquement le titre de chaque ouvrage (sans les exemplaires).

Dans un 2nd temps, définir une fonction JavaScript *affiche_exemplaires(exemplaires)* qui affiche dans la liste ** les exemplaires de l'ouvrage.

Utiliser : les mêmes références que celles proposées dans la question 2. **Recherche par auteur**

4. Recherche d'ouvrages par auteur

Dans la question 2) du paragraphe « 2. Recherche d'auteurs par nom », la fonction AJAX *recherche_ouvrages_auteur(code)* lance le programme *recherche_ouvrages_auteur.php* paramétré par *code* de l'auteur. Les données au format JSON retournées par *recherche_ouvrages_auteur.php* seront "parsées" dans la fonction *callback affiche_ouvrages(...)* pour obtenir un tableau (array) d'ouvrages. Celui-ci permettra de créer la liste des ouvrages de l'auteur de la même forme que le document vu à la « question 2) du paragraphe 3. Recherche d'ouvrages ».

Cette liste devra être attachée à la *<div>* située droite (retrouvée à partir de son *id*).

NB : faire en sorte que les affichages des listes ne se chevauchent pas.