

TD JQuery – Ajax

I) Première requête ajax

Vous disposez d'un service web à l'adresse : <http://infoweb-ens/~jacquin-c/codePostal/codePostal.php>. Ce service répond à une requête de type :

<http://infoweb-ens/~jacquin-c/codePostal/codePostal.php?commune=nantes>.

Des couples (code postal, identifiant pays) relatif à la ville ciblée sont retournés sous le format JSON (la base de données ne contient que les communes françaises).

vous avez à disposition sur madoc un fichier **saisirCommune.html** qui permet à un utilisateur de saisir un nom de commune.

- a) commencer par regarder le code JSON retourné par l'appel du web service
- b) maintenant, lorsque l'utilisateur cliquera sur le bouton *soumettre*, il faudra que si la commune est dans la base de données alors le ou les codes postaux qui lui sont associés s'afficheront dans la liste (entre les balises >/ul>). Si la commune n'est pas dans la base de données un message le stipule sur la même page.

II) Utilisation de l'UI autocomplete

Nous allons maintenant utiliser ajax et l'UI autocomplete pour réaliser de l'autocomplétion du champs de saisie de l'utilisateur. Vous avez la documentation de l'UI à l'adresse :

<http://api.jqueryui.com/autocomplete/>

Vous avez à disposition le fichier **saisirCommuneComplete.html**. Il est composé d'un champs qui permet de saisir le nom d'une commune.

- a) pour vous familiariser avec l'UI :

- faites afficher après 3 caractères saisis, une liste de suggestion qui est constituée de chaînes de caractères contenues dans une liste prédéfinie (stockée dans un tableau) que vous aurez vous même définie (il faut assigner votre tableau qui contient les chaînes prédéfinies à la propriété *source*).

- Maintenant, faites afficher dans la balise *div* de la page html, la chaîne de caractère sélectionnée. C'est la propriété *select* de l'ui autocomplete qui va être utilisée. Le deuxième paramètre de la fonction anonyme associée (ui par exemple) permet de récupérer la donnée sélectionnée. Pour ceci, il faut que le tableau (associé à *source*) contienne des objets json ayant au moins comme propriété *label* (ce qui est affiché dans la liste) et *value* (la valeur qui sera affichée dans le champs *input* par exemple après la sélection d'un item par l'utilisateur). Par exemple : [{label : " Nantes ", value: "nantes "}, ...] On peut ajouter d'autres propriétés aux objets json. On accède dans la fonction anonyme associée à la propriété *select* aux données sélectionnées par *ui.item* (exemple *ui.item.value*).

- b) Maintenant les données de la liste de suggestion seront obtenues via une requête ajax. Lorsque l'utilisateur tapera les 3 premiers caractères d'une commune, un certain nombre de noms de commune (10 par exemple) associés à des codes postaux, lui seront proposés. Lorsqu'il tapera un nouveau caractère, la suggestion sera mise à jour. Lorsqu'il sélectionnera un item dans la liste, alors le code postal s'affichera sur la même page en

dessous de la ville sélectionnée.

Pour développer ces fonctionnalités, Il faut associer à *source* une fonction anonyme : *function(request, **response**)*. Dans cette fonction, la requête ajax est envoyée au serveur. En cas de succès de celle-ci, la fonction ***response*** est appelée . Il faut lui passer en paramètre un tableau de même type que celui que vous avez manipulé précédemment (les objets json qu'il contient doivent avoir comme propriété *label* et *value* et d'autres éventuellement).

Le service web que vous utiliserez est à l'adresse : <http://infoweb-ens/~jacquinc/codePostal/codePostalComplete.php>. Il répond à des requêtes de type *GET*. Vous lui passerez en paramètre le nom de la commune et le nombre maximal de résultats retournés :

ex : commune=nantes&maxRows=100

Le résultat retourné correspond à des couples (code postal, ville).

Pour aller plus loin

Sur la même page, donnez la possibilité à l'utilisateur de faire des recherches aussi bien via les noms des communes que via les code postaux (toujours avec suggestion).