

TD4 JavaScript

EXERCICE 1 : PREMIERS PAS

Le but est d'afficher, dans un tableau HTML, une liste d'étudiants.

Pour ce faire, nous allons effectuer une requête **Ajax** afin de récupérer le fichier « **etudiants.json** ».

Chaque étudiant présent dans le fichier sera ajouté dans une ligne du tableau HTML.

Les Étudiants	
Nom	Prénom
Martin	Lucas
Bernard	Emma
Dubois	Louis
Thomas	Léa

Aide : <https://developers.google.com/web/updates/2015/03/introduction-to-fetch>

EXERCICE 2

Construire une page HTML contenant un formulaire ressemblant à ceci.

Mon formulaire

Nom	<input type="text" value="Dylan"/>
Prénom	<input type="text" value="Bob"/>
Email	<input type="text" value="bob@dylan.net"/>

Département	<input type="text" value="Choisir un département"/>
Ville	<input type="text"/>
<input type="button" value="Valider"/>	

Attention : les champs “Département” et “Ville” sont, au départ, des balises **SELECT** vides.

Première étape

Le but est de remplir le **SELECT** du département avec les données **JSON** récupérées via le **web service** dédié (requête Ajax).

Département	<input type="text" value="Choisir un département"/>
Ville	<input type="text"/>

Aveyron

Bouches-du-Rhône

Calvados

Cantal

Charente

Charente-Maritime

Cher

Corrèze

Deuxième étape

Quand on sélectionne un département, une nouvelle requête **Ajax** récupère les villes au format **JSON** via le **web service** dédié, puis les affiche dans le **SELECT** correspondant.

The screenshot shows a web form with two dropdown menus. The first dropdown, labeled 'Département', has 'Moselle' selected. The second dropdown, labeled 'Ville', has 'Aboncourt (57920)' selected, and a list of other cities is displayed below it. The cities listed are: Metting (57370), Metz (57000), Metzeresche (57920), Metzervisse (57940), Metzting (57980), Mey (57070), Mittelbronn (57370), Mittersheim (57930), and Melting (57570).

Info : le web service utilisé est - <https://geo.api.gouv.fr>

Troisième étape

Comme pour le TD3, vous devez vérifier la validité du formulaire lors d'un clic sur le bouton **Valider**.

Si le formulaire est invalide, on reste sur la page et on affiche à l'utilisateur les messages d'erreurs appropriés.

Mon formulaire

The screenshot shows a registration form with three fields: 'Nom', 'Prénom', and 'Email'. The 'Nom' field contains 'Dylan' and has an error message below it: 'Le nom est obligatoire et doit contenir entre 2 et 50 lettres.' The 'Prénom' field contains 'Bob' and has an error message below it: 'Le prénom est obligatoire et doit contenir entre 2 et 50 lettres.' The 'Email' field contains 'bob@dylan.net'.

EXERCICE 3

Créer une page qui ressemble à ceci. Elle possède un simple *INPUT*.

Rechercher une ville

Le but est de gérer une autocomplétion pour une recherche de commune.

À chaque lettre écrite dans l'input, une requête **Ajax** est envoyée vers le web service et propose à l'utilisateur les communes retournées (voir image suivante).

Rechercher une ville

met
Mettray (Indre-et-Loire - 37)
Metting (Moselle - 57)
Metzing (Moselle - 57)
Metz (Moselle - 57)
Metzeral (Haut-Rhin - 68)
Pont-de-Metz (Somme - 80)

Info : le web service utilisé est - <https://geo.api.gouv.fr>

EXERCICE 4

Construisez une page HTML ressemblant à l'image suivante.

Cette page fait des requêtes AJAX afin de récupérer le HEADER et le FOOTER.

Afficher le header

Afficher le footer

Quand on clique sur le bouton “Afficher le header”, une requête **Ajax** est lancée pour récupérer la page **header.html**. Il faut ensuite en extraire la balise DIV **#header-fragment** puis l’insérer dans la balise **header** de notre page.

Quand on clique sur le bouton “Afficher le footer”, une requête **Ajax** est lancée pour récupérer la page **footer.html**. Il faut ensuite en extraire la balise DIV **#footer-fragment** puis l’insérer dans la balise **footer** de notre page.