

BUT MMI 2ème année DW – WEB4

TP : Une table de données éditable

L'objectif de ce TP est de réviser la récupération et le traitement de données JSON, ainsi que la création, la modification d'éléments dans le DOM, la mise en place de gestionnaires d'événements. Vous mettrez en place votre projet en utilisant des modules ESM.

Membres de l'organisation

Nombre de membres : 10

Je souhaite ajouter un membre

Ajouter un membre

id	name	email
companyName	Soumettre	

	id	name	email	companyName
X	1	Leanne Graham	Sincere@april.biz	Romaguera-Crona
X	2	Ervin Howell	Shanna@melissa.tv	Deckow-Crist
X	3	Clementine Bauch	Nathan@yesenia.net	Romaguera-Jacobson
X	4	Patricia Lebsack	Julianne.OConner@kory.org	Robel-Corkery
X	5	Chelsey Dietrich	Lucio_Hettinger@annie.ca	Keebler LLC
X	6	Mrs. Dennis Schulist	Karley_Dach@jasper.info	Considine-Lockman
X	7	Kurtis Weissnat	Telly.Hoeger@billy.biz	Johns Group
X	8	Nicholas Runolfsdottir V	Sherwood@rosamond.me	Abernathy Group
X	9	Glenna Reichert	Chaim_McDermott@dana.io	Yost and Sons
X	10	Clementina DuBuque	Rey.Padberg@karina.biz	Hoeger LLC

Mise en place

- Créez un nouveau projet nommé `web4-table`, et créez le dépôt correspondant dans le groupe GitLab Unistra que vous avez créé précédemment, avec le niveau de visibilité *Private*. Initialisez le projet à l'aide de Vite à partir du template `Vanilla` comme vu dans le TP précédent, et ajoutez les packages [ESLint](#) et [JSDoc](#).
- « Nettoyez » le code de base produit par Vite : supprimez les contenus inutiles dans `index.html` et dans `main.js`, et supprimez également les fichiers

- inutiles. Dans le fichier `index.html`, ajoutez : un titre `h1` comportant le texte « Membres de l'organisation », et un élément `div` avec la valeur d'`id` `container` qui servira de conteneur pour les éléments générés par la suite.
3. En utilisant l'API Fetch, récupérez sous la forme d'un tableau d'objets JSON la table d'utilisateurs disponible à l'adresse suivante au moyen d'une requête GET :
<https://jsonplaceholder.typicode.com/users>
Il s'agit d'une API REST publique destinée aux développeurs et permettant d'effectuer des tests de requêtes GET.
 4. Ecrivez et appelez une fonction JavaScript prenant le tableau de données en entrée et générant le code HTML d'un élément `<h2>` affichant le texte « Nombre de membres : X » où « X » est le nombre de membres.
 5. Ecrivez etappelez une fonction JavaScript nommée `createTable(...)` prenant le tableau de données en entrée et générant le code HTML d'une table dont les colonnes correspondent aux propriétés suivantes d'un membre : `id`, `name`, `email`, `company.name`. Les lignes correspondront aux valeurs associées à ces propriétés pour l'ensemble des membres. Pour cela, vous pouvez utiliser la méthode `map(...)`. Vous pouvez consulter les fiches [Manipuler les propriétés d'un objet en JavaScript](#) et [Manipuler des tableaux d'objets en JavaScript](#) pour vous aider.
 6. Décomposez la fonction `createTable(...)` en plusieurs sous-fonctions afin de rendre votre code le plus lisible possible.
 7. Déplacez le code permettant de créer la table HTML dans un fichier de module séparé nommé `createTable.js`, et importez ce module dans votre `main.js`. Documentez votre module en utilisant la syntaxe [JSDoc](#).
 8. Générez la documentation du module `createTable` à l'aide de la commande :
`jsdoc src/createTable.js`. Complétez le fichier `.gitignore` avec le nom du répertoire `out` pour éviter de stocker la documentation générée dans le dépôt Git.
 9. Une limite de l'API précédente est qu'elle ne permet que des requêtes GET. Pour la suite du travail, vous allez utiliser l'[API "Users"](#), codée en PHP, à exécuter en local à l'aide de XAMPP, WAMP, Laragon etc. Cette API permet d'effectuer des requêtes GET, POST, DELETE, PUT. Pour l'utilisation, consultez le README présent avec les sources de l'API. Notez que si vous utilisez le serveur web de développement de Vite, vous aurez besoin de [configurer Apache pour les requêtes Cross-Origin](#).

0. Revenez à votre code JavaScript. Faites en sorte d'ajouter à la table HTML générée dans la question précédente une 1ère colonne avec pour chaque ligne un bouton comportant un symbole « X ». Un clic sur un de ces boutons supprimera le membre correspondant de la table HTML. Le membre devra aussi être supprimé dans le tableau de données et dans la base de données côté serveur.
1. Ajoutez une fonction JavaScript nommée `createForm(...)` générant le code HTML d'un formulaire permettant de saisir les données pour un nouveau membre, de l'ajouter au tableau de données, de mettre à jour la table HTML, et de mettre à jour la base de données côté serveur.
2. Toujours pour la même table HTML, faites en sorte qu'un clic sur une valeur de la table déclenche l'affichage d'une zone de saisie pré-remplie avec cette valeur. L'utilisateur devra pouvoir modifier la valeur et la valider en appuyant sur la touche Entrée du clavier. Une fois la validation effectuée, la zone de saisie sera remplacée par l'affichage normal du texte. Le contenu du tableau de données côté front et la base de données côté serveur devront être mis à jour suite à toute modification.