

Bases de données Locales HTML 5

Gestion d'un carnet d'adresses

Vous pouvez effectuer le TP directement dans un navigateur mais il est recommandé d'utiliser Cordova avec les smartphones de prêt (dans ce cas pensez à changer le fichier JQuery pour la version mobile).

Pré-requis : Pour des raisons de compatibilité , vous ferez le TP avec le navigateur Google Chrome ou Chromium pour les machines de la fac. Vous pouvez voir le gestionnaire de bases de données locales du navigateur en appuyant sur F12 puis l'onglet Application. Dans la section Storage vous pourrez voir les différentes données enregistrées via les API disponibles du navigateur.

I) L'API WebStorage

1) PUT

Dans cette partie vous devrez mettre en place la gestion du carnet d'adresses avec WebStorage. La mise en place est simple, la base n'a pas besoin d'être déclarée, l'insertion d'une entrée provoque la création de la base au sein du navigateur. Récupérez les données des champs du formulaire.html et créez un objet de type Entry et persistez le dans la base. Il n'y a plus qu'à remplir la méthode (en javascript) associée au bouton correspondant sur le formulaire.

Remarque : Vu que la valeur qui est associée à une clé est obligatoirement une chaîne de caractères, on vous recommande de transformer votre objet Entry en Json ... après je dis ça je dis rien.

Rappel : Pour ajouter une entrée dans la base :
`localStorage.setItem(String cle, String valeur) ;`

2) GET

Cliquez maintenant sur le lien « Consulter le carnet ». Le but est maintenant de récupérer une de vos entrées avec la clé que vous aurez choisi

et de l'afficher dans les champs correspondants. Remplissez la méthode associée à la suppression d'une entrée en javascript.

Remarque : ... Json.

Rappel : Pour récupérer une entrée dans la base :
`localStorage.getItem(String cle) ;` => retourne une chaîne de caractères.

3) DELETE

Maintenant vous devez supprimer une de vos entrées. Servez-vous encore de la clé que vous aurez choisie et remplissez la méthode en javascript permettant de supprimer une entrée.

Rappel : Pour supprimer une entrée de la base :
`localStorage.removeItem(String cle) ;`

4) Facultatif – Supprimer la base

Pour supprimer la base, vous avez juste à remplir la méthode associée au bouton dans le fichier javascript.

Rappel : `localStorage.clear() ;`

II) L'API WebSQL

L'objectif de cette partie va être d'initialiser une BDD, créer une table et faire quelques opérations SQL dessus. Le code HTML vous à été donné, plus qu'à remplir les trous !

1) Création de la base de données :

Rien de bien méchant, il suffit de déclarer une variable comme suit :

```
db = openDatabase(NOM ,VERSION , DESCRIPTION , TAILLE_ESTIMEE);
```

Il faut faire cette opération dans le « `if(window.openDatabase)` ».

Vous pouvez rajouter un cinquième argument qui sera une fonction appelée si la base est bien créée.

2) Création de la table Personne :

Ouvrez une transaction depuis votre BDD puis faites la requête SQL dans celle-ci pour créer la table Personne. Pour ce faire les deux commandes suivantes sont à imbriquer :

```
db.transaction(function(tx){FONCTION SQL}, FONCTION_ERROR, FONCTION_SUCCESS);
```

```
tx.executeSql('CREATE TABLE nom_de_la_table (arg1, arg2, ..)',  
[vide],FONCTION_ERROR, FONCTION_SUCCESS);
```

note : n'oubliez pas de créer les fonctions appelées en cas d'erreur ou de succès, une alerte est toujours utile pour savoir si la requête a fonctionné.

3) Ajout d'une personne dans la BDD:

La même chose que précédemment, mais en différent ! Il faudra passer des arguments par l'intermédiaire du tableau que l'on a laissé vide lors du CREATE :

```
tx.executeSql('INSERT INTO table (arg1,arg2,...) VALUES  
(?, ?, ...)', [arg1,arg2,...], error, success);
```

note : F12 pour avoir les informations sur les BDD, voir si la personne a bien été créée.

4) Visualisation d'une personne dans la BDD:

Même méthode que précédemment mais avec un select, la encore la variable dans la requête sera remplacée par un « ? » et mis dans le tableau de la fonction. Le résultat sera envoyé dans la fonction success de executeSql(). Nous l'appellerons DisplayResults. Elle aura cette forme :

```
function displayResults( tx, results )
```

results sera sous la forme d'un rows, l'équivalent d'un tableau d'éléments. Chaque élément sera ici une personne.

results.rows.item(i).nom renverra l'attribut nom de la personne à la position i du rows.

Afficher cette personne dans le html.

5) A vous de jouer ! :

A partir de maintenant vous avez tout les éléments en main. Supprimez une puis plusieurs personnes. Ensuite vous pouvez vous « amuser » à créer d'autres tables pour les lier entre elles.