

TP SQL - Utilisation des PHP PDO avec JavaScript (FETCH)

JS{FETCH} est une technique qui permet de faire une requête à un serveur en arrière plan, sans rechargement de la page principale. C'est le remplaçant de la technologie Ajax.

FETCH fonctionne sur le principe des PROMESSES (Alors qu'Ajax fonctionne sur le principe des « événements »)

Exemple : Appeler la page page1.php en arrière plan. Cette page renvoie du texte. On affiche le texte :

```
fetch("page1.php ") // Accès Fetch au php sans recharger la page html
.then( reponseTexte => { info.innerHTML = reponseTexte ;})
.catch ( reponseTexte => { console.log(reponseTexte) ; }) ;
```

.then et .catch sont les retours de la promesse fetch.

.then s'exécute quand le fetch a terminé son travail. La variable reponseTexte contiendra ce que la page1.php a envoyé.

.catch renvoie le message d'erreur généré par fetch dans le cas où il n'a pas pu atteindre page1.php.

Votre environnement de travail :

1. Copier les fichiers sur votre espace « Serveur WEB » sur le serveur du Labo 10.69.88.1(avec Filezilla).
2. Avec un navigateur, vérifiez le fonctionnement du fichier PHP locavelo.php. Vous devriez obtenir une liste JSON de marque de vélo.
3. Avec HeidiSQL, observez la structure de la base de données complète.
4. Testez la page locavelo.html pour vérifier que vous avez la même liste dans le SELECT.
5. Créer 2 autres fichiers PHP pour obtenir la liste des « typesVelo » et « taillesVelo ».
6. Modifier locavelo.html pour y préparer l'arrivée des 2 listes supplémentaires
7. Modifier locavelo.js pour récupérer les 2 listes supplémentaires à partir des fichiers PHP et de requêtes Fetch
8. Afficher la liste des vélos du stock en fonction des type, marque, taille choisis dans les selects.