

# Exercice 3 - Requêtes asynchrones (sommatif)

## Devis (5)

Nous avons vu ces trois comportements en classe (ajout (.5), modification (.5) et suppression (.5)) pour assurer la gestion des équipes associées à la base de données `ligue-async` de façon asynchrone (AJAX ou *Fetch*). L'exercice consiste à regrouper ces trois fonctionnalités comme si c'était une page côté administrateur. Plus précisément, les consignes sont :

- Le fichier `index.php`, les styles, le `SQL` et les instructions `PHP` sont déjà donnés, il vous reste à écrire les comportements `JavaScript`.
- Côté `PHP`, remarquez que le format de requête attendu est `JSON`.
- (.5) Les fichiers `JS` sont déjà créés. Vous devez travailler en `POO` de type module. N'oubliez pas les `#`. Et les fonctions fléchées sont interdites.
- Il n'y a pas de validation à faire pour cet exercice.
- (1) Les actions de l'utilisateur seront directement injectées, c'est-à-dire :
  - à l'ajout d'une équipe, celle-ci est ajoutée à la liste (injection `DOM`) et pensez à nettoyer les champs du formulaire ;
  - au clic d'un bouton 'Changer', le nouveau nom de l'équipe est injecté et pensez à nettoyer le champ du formulaire ;
  - au clic d'un bouton 'Supprimer', cette équipe est supprimée du `DOM`.
- (2) Les comportements de toutes les équipes affichées doivent être fonctionnels en tout temps sans rechargement de page. Évidemment, si l'utilisateur recharge la page, il devrait retrouver exactement la même liste d'équipes - et leurs comportements - affichée avant le rechargement.
- Petit truc, pour synchroniser les comportements d'une nouvelle équipe, assurez-vous d'injecter son `ID` retourné par la fonction `ajouteEquipe()`.
- Pour des points bonus virtuels, rassemblez tous les comportements serveurs (`ajouteEquipe`, `changeNomEquipe` et `supprimeEquipe`) à l'intérieur du fichier `requetesAsync.php`.