



## Module 2 Développer pages web en lien avec base de données

Programmation Client/server  
Side

Séance

S03

Activité

AJ-001

Objectif de cette séance :

Comprendre les mécanismes de traitement asynchrone et de rafraichissement partiel de page proposés dans AJAX et, dans le cadre de cet atelier, ceux de l'objet de base sur lequel s'appuie ces techniques, XMLHttpRequest.

Au cours de cet atelier, vous devrez mettre en place des échanges asynchrones qui traitent de requêtes fournissant des données dans le format Json

Sommaire de l'activité proposée :

1. Mettre en œuvre des traitements asynchrones .....2

# 1. Mettre en œuvre des traitements asynchrones

## Travail à réaliser :

A partir des éléments fournis dans le support de cours sur l'objet XMLHttpRequest.

Vous allez construire une application qui effectue des appels asynchrones d'api Web, ici celles exposés par le site JsonPlaceholder <https://jsonplaceholder.typicode.com/>

Programmez plusieurs méthodes pour extraire les données depuis la méthode de base avec XMLHttpRequest, une parmi les nombreuses méthodes du composant Ajax de JQuery et enfin, la plus simple et la plus moderne, avec la méthode fetch de javascript.

Vous devez afficher la liste des utilisateurs au sein d'un tableau HTML qui se construira dynamiquement à partir des résultats obtenus auprès du service de JsonPlaceholder. Affichez les informations nom, email, site web.

Pour la gestion, des éléments en tableau, je vous suggère de mettre en œuvre les composants de la bibliothèque Ajax JQuery Datatable [https://datatables.net/examples/data\\_sources/ajax](https://datatables.net/examples/data_sources/ajax).

Un composant très bien fait qui permet de charger des données, les paginer et les trier au sein d'une table.

Sur sélection d'un des utilisateurs du tableau, affichez ses articles au sein d'une autre table.

Pour obtenir la liste de ces articles, utiliser la méthode GET appropriée parmi celles proposées ci-dessous

Routes	
All HTTP methods are supported.	
GET	<a href="#">/posts</a>
GET	<a href="#">/posts/1</a>
GET	<a href="#">/posts/1/comments</a>
GET	<a href="#">/comments?postId=1</a>
GET	<a href="#">/posts?userId=1</a>
POST	<a href="#">/posts</a>
PUT	<a href="#">/posts/1</a>
PATCH	<a href="#">/posts/1</a>
DELETE	<a href="#">/posts/1</a>