Chapitre 7:

Http, Ajax, JSON

JavaScript



Email: amar@jant.tech



Plan

- •HTTP
- •JSON vs XML
- •AJAX
 - Fonctionnement
 - Exemple
 - TP
- ·Fetch API
 - Fonctionnement
 - Exemple
 - TP

HTTP

HTTP (Hypertext Transfer Protocol) est l'ensemble de règles régissant le **transfert de données** sur le Web.



JSON vs XML

```
"name": "John",
    "age": 30,
    "city": "New York"
}
```

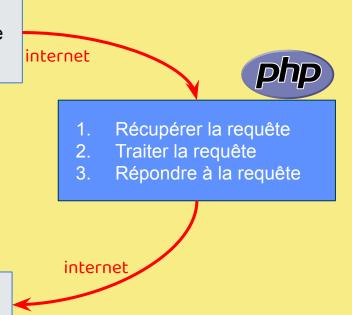
Ajax

Ajax signifie "Asynchronous JavaScript and XML" (JavaScript et XML asynchrones). Il s'agit d'une technique pour créer des pages Web rapides et dynamiques qui mettent à jour le contenu sans nécessiter un rechargement complet de la page. Ajax utilise une combinaison de HTML, CSS, JavaScript et XMLHttpRequest (XHR) pour envoyer et recevoir des données de manière asynchrone entre le navigateur Web et le serveur.

Fonctionnement 1/2



- 1. Création de la requête
- 2. Paramètre de la requête
- 3. Ecoutez la réponse



JavaScript

Traiter la réponse

Fonctionnement 2/2

- 1. Un événement se produit dans une page Web
- 2. Un objet XMLHttpRequest est créé par JavaScript
- 3. L'objet XMLHttpRequest **envoie une requête** à un **serveur Web**
- 4. Le serveur traite la requête
- 5. Le serveur **renvoie une réponse** à la page Web
- 6. La réponse est lue et ou transformée par JavaScript
- 7. L'**action** appropriée (comme la mise à jour de la page) est **effectuée** par JavaScript

Ajax: Exemple de récupération de données JSON

```
function get agents ajax() {
  var xhr = new XMLHttpRequest();
  xhr.open('GET', 'http://localhost/uahb/projet1/backend/index.php);
  // écouteur de la réponse
  xhr.onload = function () {
      if (xhr.status === 200) {
           let response = xhr.responseText
           let donneeJson = JSON.parse(response)
           console.log(donneeJson);
       } else {
           console.log('Erreur : code=' + xhr.status);
   };
  xhr.send();
get agents ajax()
```

Exemple backend avec PHP

```
http://localhost/uahb/projet1/backend/index.php
// accepter toutes requêtes
header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Headers: *");
header ('Access-Control-Allow-Credentials: true');
header("Access-Control-Allow-Methods: *");
// connexion de la base de données
$connexion = new
PDO ("mysql:host=localhost;port=3306;dbname=ma bd;", "root", "root");
// récupération des données à partir de la base de données
$lesFournisseurs = $connexion->query("select * from fournisseur")
                               ->fetchAll(PDO::FETCH ASSOC);
//Envoyer la réponse sous format JSON
echo json encode($lesFournisseurs);
```

TP 1

- 1. Créez une base de données **l2_bd**
 - a. créez la table etudiant(id_etudiant, prenom, nom, age)
 - b. Insérez 3 étudiants
- 2. Créez une projet PHP
 - a. créez un fichier **list_etudiant.php** permettant de renvoyer la liste des étudiants sous **format JSON**
- 3. Créez un projet JavaScript
 - a. Avec un fichier **main.js** qui récupère la liste des étudiants en utilisant **AJAX**
 - b. Créer un fichier **index.html** ou vous allez afficher la liste de manière ordonnée (en utilisant les balises ul, li)

Fetch API

L'API Fetch est une **récente fonctionnalité** de JavaScript permettant d'effectuer des requêtes **HTTP**.

Elle est supportée par la plupart des navigateurs modernes.

Fetch API: Exemple frontend

```
function get agents() {
   fetch('http://localhost/uahb/projet1/backend/index.php)
       // transformation de la réponse sous format JSON
       .then(response => response.json())
       // Traitement après réponse
       .then((data) \Rightarrow
           console.log(data)
       // en cas d'erreur
       .catch(error => console.error(error));
get agents()
```

TP 2

Réalisez le TP1 en utilisant Fetch API

Ajax: Exemple d'envoie de données

```
function inscription()
let input email = document.getElementById("id email")
let input password = document.getElementById ("id password")
var xhr = new XMLHttpRequest();
xhr.open('POST', 'http://localhost/uahb/projet1/backend/inscription.php' , true);
xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
xhr.onload = function () {
  if (xhr.status === 200) {
    let response = xhr.responseText
    let donneeJson = JSON.parse(response)
    console.log(donneeJson);
    console.log('Erreur : code=' + xhr.status);
xhr.send("email=" + encodeURIComponent(input email.value) + "&password=" +
encodeURIComponent (input password.value));
```

Fin de JavaScript Chapitre