# Ajax

## Définition et composants

Ajax ou Asynchronous JavaScript And XML.

Est un ensemble de techniques permettant de créer des applications web plus interactives, en actualisant dynamiquement une partie d'une page sans la recharger entièrement.

Ajax est composé d'un Assemblage de produits et techniques qui sont :

1. HTML + CSS : pour la structure et la présentation de la page.
2. JavaScript : pour les traitements côté client.
3. XML (de plus en plus remplacé par JSON) : pour échanger des données entre le client et le serveur.
4. XMLHttpRequest (XHR) : pour envoyer et recevoir des données en mode asynchrone (middleware entre le client et le serveur).

Ajax introduit une distribution des traitements entre client et serveur, participant au mouvement du Web 2.0 et favorisant l’émergence d’applications web plus dynamiques.

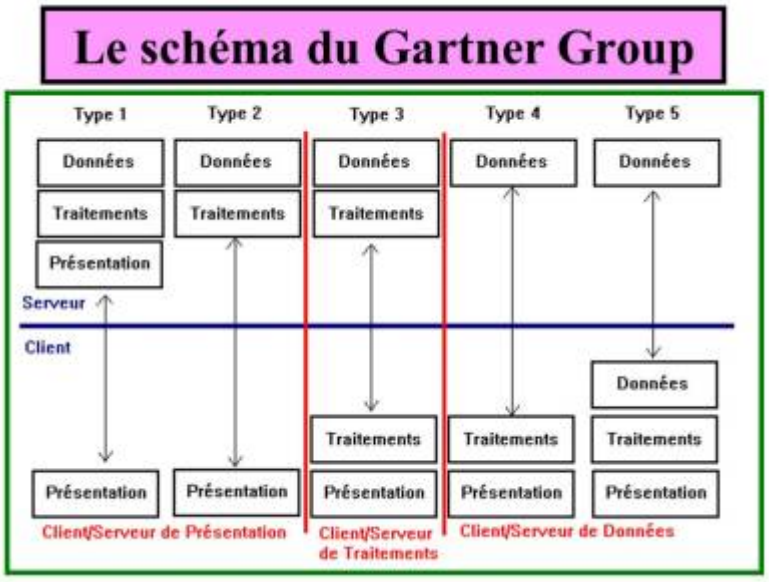
## Spécificité d'architecture

Ajax modifie l'architecture classique client-serveur :

Certaines logiques de traitement sont déplacées du serveur vers le client.

Le modèle architectural proposé suit les classifications du Gartner Group, où les traitements sont mieux distribués et les interactions plus riches.

L'utilisation d'Ajax permet ainsi d'alléger les serveurs et de rendre les applications plus réactives sans solliciter constamment des rechargements complets de page.




# Principe de fonctionnement

Ajax repose sur une communication asynchrone entre le client et le serveur :

Lorsqu'une interaction utilisateur se produit (clic, saisie, etc.), un objet XMLHttpRequest est créé côté client.

Cet objet envoie une requête HTTP en arrière-plan vers le serveur sans interrompre l'affichage ou l'interaction en cours.

Le serveur traite la demande et renvoie une réponse (souvent au format JSON ou XML).

JavaScript traite ensuite cette réponse pour mettre à jour dynamiquement la page.

Un schéma de principe illustre cette chaîne : Interaction utilisateur → Envoi d'une requête XHR → Réponse du serveur → Mise à jour dynamique de la page.

---

En Ajax, les requêtes sont asynchrones, c'est-à-dire qu'elles ne bloquent pas le reste de l'application pendant leur exécution. Lorsqu'une requête est envoyée :

Une fonction de rappel (callback) est définie pour être exécutée dès que la réponse du serveur est disponible.

Cette approche permet d'améliorer la fluidité de l'expérience utilisateur et d'éviter les temps d'attente perceptibles.

Exemple simple en JavaScript:

```
    var xhr = new XMLHttpRequest();
    xhr.open('GET', 'data.json', true);
    xhr.onload = function() {
        if (xhr.status === 200) {
```

```
            console.log(JSON.parse(xhr.responseText));
        }
    };
    xhr.send();
```

## Avantages

Les principaux avantages d'Ajax sont :

- Réactivité accrue des applications web.

- Réduction du trafic réseau, car seules les données nécessaires sont échangées.

- Soulagement des serveurs, certains traitements étant déportés vers le client.

- Faible consommation de ressources côté client pour des interactions rapides et légères.

## Inconvénients

Malgré ses avantages, Ajax présente aussi des limites :

- Nécessité d'activer JavaScript côté client pour son fonctionnement.

- Compatibilité navigateur : tester sur différents navigateurs est indispensable.

- Problèmes de référencement : les moteurs de recherche peuvent avoir du mal à indexer des contenus chargés dynamiquement.

- Risques de sécurité, notamment d'injection de code via les requêtes.

- Temps de chargement initial parfois plus long dû aux bibliothèques JavaScript nécessaires.

- Détournement possible de l'utilisation de XmlHTTPRequest pour des usages malveillants.

From:
https://wiki.siochaptalqper.fr/ - **Wiki SIO Chaptal**

Permanent link:
**https://wiki.siochaptalqper.fr/doku.php?id=bloc2:prog:web:ajax&rev=1745842272**

Last update: **2025/04/28 14:11**