

# Avant-propos

---

Popularisé par des sites innovants tels que Google Suggest, Google Maps, les webmails de Google et Yahoo, writely, iRows ou netvibes, Ajax (Asynchronous JavaScript And XML) est un ensemble de technologies permettant de construire des applications Web de nouvelle génération, comparables en fonctionnalités aux applications natives Windows ou Mac OS.

Ajax associe le HTML dynamique, qui permet de modifier le contenu ou l'apparence de la page localement (sans redemander une nouvelle page au serveur), avec des appels asynchrones au serveur, qui récupèrent juste les fragments à mettre à jour. Ces technologies existant depuis plusieurs années, Ajax est disponible aujourd'hui dans les navigateurs installés, tels Internet Explorer 6, Firefox 1.x, Safari 1.3, Opera 8.5, etc.

Ajax rend les applications Web plus réactives et leurs interfaces utilisateur plus riches.

La réactivité tient à trois facteurs :

- Fluidité : les échanges avec le serveur sont à la fois plus fréquents et d'un volume plus réduit, ce qui « lisse » la mise à jour dans le temps. De plus, ces mises à jour étant asynchrones, elles peuvent se dérouler en parallèle et ne bloquent pas l'utilisateur.
- Rapidité : les échanges étant d'un volume plus réduit, le transfert est plus rapide, de même que le traitement local pour mettre à jour l'affichage.
- Sensibilité : les applications peuvent réagir à plus d'événements, notamment à ceux liés à la saisie clavier.

La richesse d'interface tient aussi à trois facteurs :

- Richesse des composants graphiques : en plus des formulaires et des liens, qui sont, en HTML, les principaux éléments réactifs, il est possible de disposer d'onglets, de boîtes flottantes, d'arborescences, de menus, d'info-bulles, de calendriers, etc.
- Support du glisser-déposer et, dans une certaine mesure, de mécanismes de défaire/refaire.
- Support de l'édition WYSIWYG.

Les applications Ajax combinent ainsi la puissance et la qualité de l'interface utilisateur du client lourd avec l'absence de déploiement du client léger, constituant par là ce qu'on appelle aujourd'hui le client riche.

## Objectifs de l'ouvrage

Cet ouvrage vise à guider le lecteur dans la réalisation d'applications Ajax. Il couvre trois grands axes : les technologies constitutives d'Ajax, les questions fonctionnelles et techniques soulevées, avec les réponses qu'il est possible d'y apporter, et enfin les méthodes et outils qui facilitent et rendent plus sûr le développement.

Les technologies comportent :

- Le HTML dynamique, c'est-à-dire DOM (Document Object Model), ou modèle objet du document, et les CSS (Cascading Style Sheets), ou feuilles de style en cascade, combinés avec JavaScript.
- XMLHttpRequest, l'objet JavaScript permettant d'effectuer des requêtes HTTP asynchrones vers le serveur.
- JSON (JavaScript Object Notation), ou notation objet en JavaScript, ainsi que XML et les technologies associées, notamment XSLT et XPath.

Les questions soulevées concernent la gestion des appels asynchrones, la conception et l'utilisation de composants graphiques (comme la suggestion de saisie) ou fonctionnels (comme un panier dans une application de vente en ligne), la complexité du code JavaScript induite par la richesse des fonctionnalités du client et l'intégration d'Ajax dans les applications Web existantes.

Les solutions à ces problèmes reposent en grande partie sur des méthodes et des frameworks : pour faire face à la complexité du code, celui-ci doit être conçu de façon modulaire, en faisant appel à la programmation objet de JavaScript, à des bibliothèques de composants JavaScript et à une architecture MVC (modèle, vue, contrôleur).

Cette dernière, qui a démontré sa valeur côté serveur ainsi que dans les applications à client lourd, se révèle désormais utile aussi côté client en Web. Elle se prolonge par la séparation nette entre le code HTML, concentré sur la structure et le contenu, le code JavaScript en charge du comportement, et les CSS, qui gouvernent l'apparence.

Les méthodes valant avant tout par l'exemple (« faites ce que je fais » étant en cette matière plus efficace que « faites ce que je dis »), elles sont constamment mises en pratique dans cet ouvrage.

L'objectif de celui-ci est en effet double : installer chez le lecteur des bases solides et le rendre opérationnel rapidement à travers des réalisations d'ampleur progressive, le dernier chapitre incluant une étude de cas illustrant comment concevoir et réaliser une application « ajaxifiée ».

## Organisation de l'ouvrage

Dans chaque chapitre, l'ouvrage part du besoin, l'objectif à atteindre, et guide progressivement le lecteur vers cet objectif. La succession des chapitres vise également à faciliter cette progression. L'explication des techniques est illustrée par des exemples, que le

lecteur pourra, dans certains cas, réutiliser dans ses propres projets. L'auteur insiste sur les fondements de chaque technologie et les pratiques conseillées et indique les points délicats, facteurs de bogues fréquents.

Le chapitre 1 examine ce qu'apporte Ajax et liste notamment les cas typiques où il offre un réel avantage. Une grande partie des exemples présentés ici sont développés dans les chapitres suivants. Les fonctions et le code d'un exemple simple sont examinés en détail afin de montrer comment s'imbriquent les différentes composantes du code.

Il n'est pas possible de faire de l'Ajax sans faire du HTML dynamique, qui consiste à modifier le contenu, la structure ou l'apparence d'une page Web sans aucun appel au serveur. La réciproque est fautive. Aussi est-il plus facile de commencer par le HTML dynamique, auquel le chapitre 2 est consacré. Ce chapitre fait un point sur les feuilles de style CSS et sur l'API DOM, qui permet de manipuler la structure, le contenu et l'apparence de la page Web et de gérer les événements liés aux actions de l'utilisateur.

Les applications Ajax faisant massivement appel à des composants, le chapitre 3 se penche sur les aspects objet et avancés de JavaScript, qui sont très particuliers et peuvent être déroutants tant pour les développeurs Java ou C# que pour ceux qui n'ont pas une grande habitude de la programmation objet. Le chapitre est illustré par la création de composants graphiques et d'une suggestion de saisie en local (comme sur le site de la RATP). Ce chapitre est fondamental, car tous les exemples des chapitres suivants construisent (ou font appel à) des composants JavaScript.

Le chapitre 4 est dédié à la communication avec le serveur *via* des requêtes XMLHttpRequest. Après un point sur le protocole HTTP, il examine les questions soulevées par les appels Ajax : gestion du cache, méthodes GET et POST, requêtes parallèles, cookies, sécurité. Les exemples incluent une suggestion de saisie par appels au serveur et la mise à jour dynamique d'une liste déroulante, réalisées sous forme de composants.

Le chapitre 5 traite de l'échange de données XML ou JSON entre le client et le serveur. Il examine et compare les différents formats d'échanges (texte, HTML, XML et JSON). Les spécificités de XML et de son DOM par rapport au HTML y sont indiquées. Il traite aussi de XSLT (transformations XML) et XPath. Les technologies sont illustrées par un lecteur RSS, d'abord dans une version simple faisant appel à DOM, puis dans une version plus élaborée faisant appel à XSLT.

Le chapitre 6 aborde les « frameworks » Ajax. Il commence par différencier bibliothèques et frameworks, puis examine les critères de choix à prendre en compte et compare les produits disponibles. Il étudie en particulier dojo et prototype.js, les deux bibliothèques les plus anciennes et, semble-t-il, les plus utilisées actuellement. Le chapitre est illustré de nombreux exemples d'utilisation, notamment une mise en œuvre du glisser-déposer et un éditeur WYSIWYG.

Le chapitre 7 et dernier est consacré aux applications Ajax et Web 2.0. C'est le plus important du point de vue de la conception et de l'architecture. Il examine les questions soulevées par Ajax, en particulier l'absence de support des actions Page précédente et Ajouter aux favoris. Il détaille l'impact d'Ajax sur les architectures Web, en particulier le

modèle MVC. Une étude de cas conclut le chapitre et l'ouvrage en montrant qu'au MVC côté serveur s'adjoint dans les cas complexes un MVC côté client.

Chaque chapitre s'appuie sur les connaissances acquises au cours des chapitres précédents et les enrichit. À l'issue de ce parcours, le lecteur aura un bagage solide et disposera de composants directement utilisables, lui permettant d'aborder ses projets Ajax en étant opérationnel.