

# Introduction

La technologie AJAX (acronyme d'*Asynchronous JavaScript and XML*) permet de modifier une partie de la page Web par la transmission en arrière-plan d'un fichier qui vient s'insérer dans la page d'origine sans avoir recours au rechargement de la page complète. Ce processus est totalement transparent pour l'utilisateur. Celui-ci ne peut que constater que l'application est interactive, fluide et ergonomique. Toutes ces qualités de navigation s'intègrent dans le concept des applications Web 2.0.

Apparu en 2005, AJAX a depuis été largement adopté par les développeurs, spécialement ceux qui ont en charge des applications dont le contenu connaît de nombreuses mises à jour. Si certains qualifiaient à l'époque AJAX d'effet de mode, ils doivent bien admettre maintenant que la technologie AJAX est toujours adoptée de nos jours par les concepteurs et ce de façon croissante.

La mise en œuvre d'AJAX fait appel à plusieurs composants de la publication sur la toile soit le JavaScript, le XML, le DOM (*Document Object Model*) et l'API XMLHttpRequest. Ainsi, son apprentissage peut, au premier abord, paraître complexe, mais en connaissant les tenants et aboutissants de chaque élément la mise en place d'une requête AJAX se révèle finalement chose relativement aisée. C'est là tout notre objectif lors de l'écriture de ce livre ! Nous consacrons un chapitre à chaque composant d'AJAX. Un premier chapitre est consacré au JavaScript qui gère l'ensemble du processus et initiera la requête vers le serveur. C'est à ce moment qu'intervient l'API XMLHttpRequest qui effectue la requête de façon asynchrone soit en arrière-plan de la page, sans perturber le flux normal de celle-ci. La réponse du serveur à la requête s'effectue théoriquement en XML qui structure ainsi les données transmises. Ces dernières prendront place dans l'arbre du DOM du document initial, procédant ainsi à l'affichage des nouvelles données sans avoir à recharger l'entièreté du document.

Il ne faut cependant pas se bloquer sur le XML du XMLHttpRequest ou du X d'AJAX car nous verrons que ces données peuvent être en texte brut, en HTML et même en JavaScript.

Les différents éléments décortiqués, l'ouvrage regroupe le tout dans le chapitre consacré à l'approche AJAX pour récupérer du texte, du HTML et du XML. Pour illustrer le processus AJAX celui-ci est alors repris dans une dizaine d'exemples. Ces derniers reprennent, dans des applications variées, les éléments étudiés plus avant dans le livre avec le code commenté ligne par ligne.

Dans cette deuxième édition, nous ne pouvions ignorer l'apparition des frameworks JavaScript et en particulier le plus populaire d'entre eux qu'est jQuery. Après avoir présenté quelques-unes de ses spécificités, comme les sélecteurs, la gestion des événements, les effets visuels ainsi que la manipulation des feuilles de style CSS et du DOM, nous abordons un chapitre présentant AJAX sous jQuery. Avec une instruction raccourcie, tout le processus XMLHttpRequest est repris en une seule ligne de code, simplifiant ainsi grandement le travail des développeurs. jQuery introduit également une série d'événements qui permettent de suivre pas à pas le déroulement de la requête. Il autorise aussi d'introduire aisément une icône de chargement lorsque ce dernier se révèle plus long que prévu. Enfin, de nouveaux exemples AJAX mettent un point final à ce chapitre consacré à jQuery ainsi qu'à cet ouvrage.

# Principe et définition

Le terme AJAX est apparu pour la première fois le 18 février 2005, dans un essai de James Garret intitulé "Ajax : une nouvelle approche pour les applications Web".

AJAX est l'acronyme d'*Asynchronous JavaScript and XML* (JavaScript et XML asynchrone) qui désigne une approche innovante dans la conception de pages web dont l'objectif est d'optimiser leur interactivité et leur confort d'utilisation.

AJAX n'est pas une technologie nouvelle, c'est un terme synthétique qui désigne l'utilisation conjointe de différentes technologies dans la création de pages web. Ainsi AJAX incorpore :

- le HTML et les feuilles de style CSS ;
- le JavaScript ;
- le DOM ;
- l'objet XMLHttpRequest ;
- le XML et le XSL.



Le HTML et les feuilles de style CSS prennent en charge la présentation des pages de façon standardisée.

Le JavaScript, qui marque ainsi son grand retour dans l'univers de la publication sur le web, est omniprésent dans les applications AJAX.

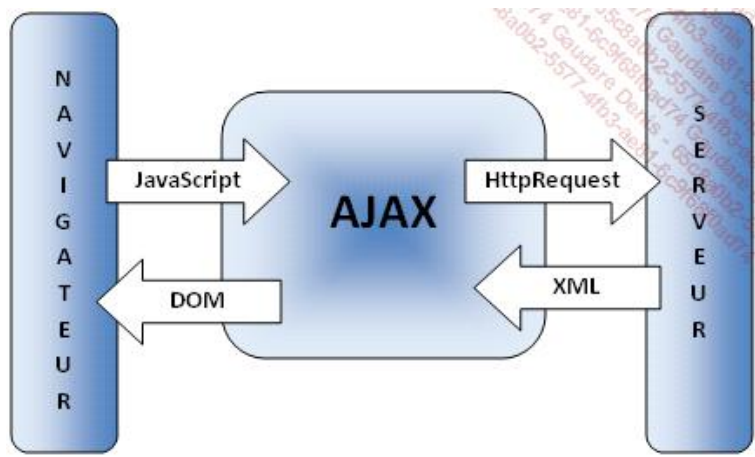
Les objets du document sont pris en charge selon le DOM du consortium W3C (*World Wide Web Consortium*).

L'objet XMLHttpRequest permet de lancer des requêtes de la page web vers le serveur pour y récupérer des fichiers. Son fonctionnement permet d'effectuer ces requêtes de façon asynchrone, soit en arrière-plan de la page et de façon complètement transparente pour l'utilisateur.

Les fichiers récupérés peuvent être au format XML ; les apports de ce format dans le domaine de l'échange des données ne sont plus à présenter.

Il s'agit donc de techniques, du JavaScript au XML en passant par le DOM et l'objet XMLHttpRequest, qui sont éprouvées, standardisées et maîtrisées. Ce qui est assez prometteur quant à la compatibilité d'AJAX avec les navigateurs actuellement utilisés sur le Web.

Mais c'est dans l'utilisation conjointe de ces différentes technologies que réside l'originalité du fonctionnement d'AJAX, que nous appellerons par la suite le modèle AJAX.

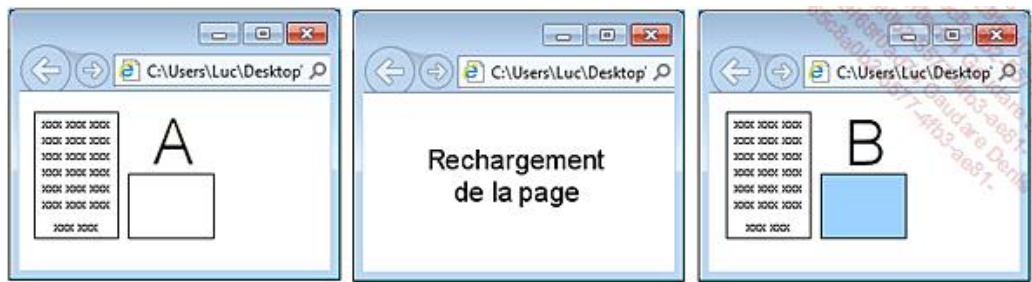


Ce modèle AJAX se décompose comme suit :

- La page web s'affiche, comme à l'accoutumée, dans le navigateur, appelé aussi le client.
- La moindre interaction de l'utilisateur, par exemple l'encodage d'un formulaire ou le cliqué/déplacé d'un élément, est prise en charge par la gestion des événements de JavaScript.
- Le code JavaScript de la page initie une requête `XMLHttpRequest` vers un fichier XML situé sur le serveur.
- Le fichier XML réclamé est renvoyé vers le navigateur de l'utilisateur.
- Il est alors pris en charge par le DOM, et toujours par le JavaScript, pour être traité et affiché de façon dynamique dans la page initiale.

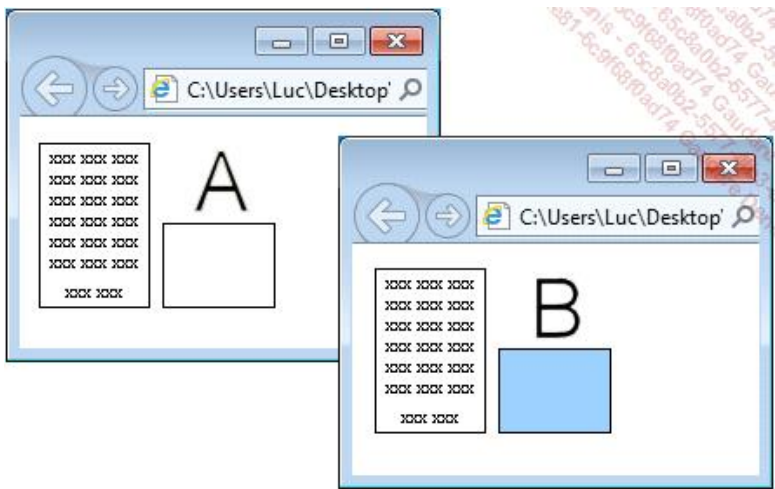
Toute cette procédure s'est déroulée de façon transparente pour l'utilisateur, sans nécessiter le fastidieux rechargement de la page.

Le modèle AJAX permet ainsi de sortir du schéma traditionnel de la conception de pages web dans lequel chaque nouvelle information entraîne l'affichage d'une nouvelle page.



Avec AJAX, une nouvelle information peut être récupérée dans un petit fichier situé sur le serveur. Ce fragment d'information peut alors être pris en charge et affiché dans une zone de la page et compléter ainsi la page initiale.

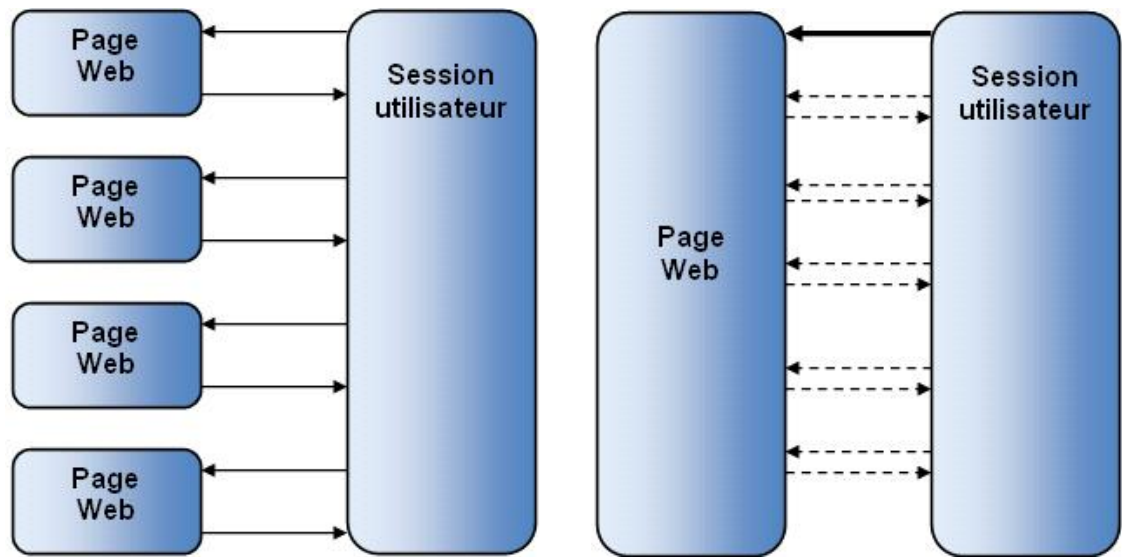
En évitant l'actualisation de la page et la perte de temps que cela entraîne, les applications gagnent ainsi en fluidité. En outre l'ergonomie s'en trouve améliorée car l'utilisateur peut rester concentré sur la lecture de la page ou la tâche accomplie.



Le schéma traditionnel de la relation client-serveur, mis en place dès les origines du Web par la caractéristique hypertexte des pages web, est ainsi mis en question. Selon celui-ci, la plupart des interactions de l'utilisateur comme le clic sur un lien ou l'envoi d'un formulaire, déclenchent une requête HTTP vers le serveur web. Ce dernier, après le traitement éventuel des données, retourne une nouvelle page HTML ou XHTML au client.

La session de l'utilisateur est alors une succession de requêtes HTTP et de réponses du serveur. Cet aller-retour incessant du client vers le serveur affiche à chaque fois une nouvelle page dans le navigateur.

Avec le schéma AJAX, la page web initiale comprend une part importante de code AJAX qui permet, soit de prendre en charge une série d'opérations comme la validation de formulaires, soit d'initier des requêtes XMLHttpRequest pour toutes informations complémentaires nécessaires. Les données ainsi retournées viennent s'incorporer de façon asynchrone dans la page initiale en apportant une facilité d'utilisation et une réactivité jusqu'alors inconnues sur la toile.



Le schéma traditionnel

Le schéma AJAX

La puissance du concept AJAX est telle que l'on peut envisager des applications web qui se rapprochent des applications logicielles, appelées aussi applications de bureau. Ce qui modifie complètement la richesse de l'interface et les possibilités offertes par les services des pages web.

Mais l'enjeu d'AJAX va plus loin que ces considérations techniques. Son fonctionnement réactif permet de reconsidérer totalement l'ergonomie des sites web. Alors que pendant près d'une décennie, la réalisation de sites plaisants à l'œil était d'actualité, à présent la conception de sites plaisants à être utilisés est envisageable. L'utilisateur devient ainsi l'élément central des préoccupations des développeurs.

La définition d'AJAX n'est cependant pas à prendre à la lettre. L'approche AJAX est dans la pratique plus souple. Soulignons par exemple que l'objet `XMLHttpRequest` permet de récupérer non seulement des fichiers XML, mais aussi de simples fichiers de texte (au sens informatique du terme). Cette caractéristique élargit considérablement le champ des possibilités offertes aux développeurs.

Le concept AJAX est mis en avant et soutenu par Google. De nombreuses applications signées Google en reprennent l'approche. On songe ainsi à Google Maps et Google Suggest que nous allons aborder au point suivant. Il met également à la disposition des concepteurs un kit de développement (en open source). Le fait qu'une figure emblématique du Web comme Google s'investisse ainsi dans les applications AJAX ne peut que susciter ou renforcer l'intérêt des développeurs de sites web.

# Exemples sur le Web

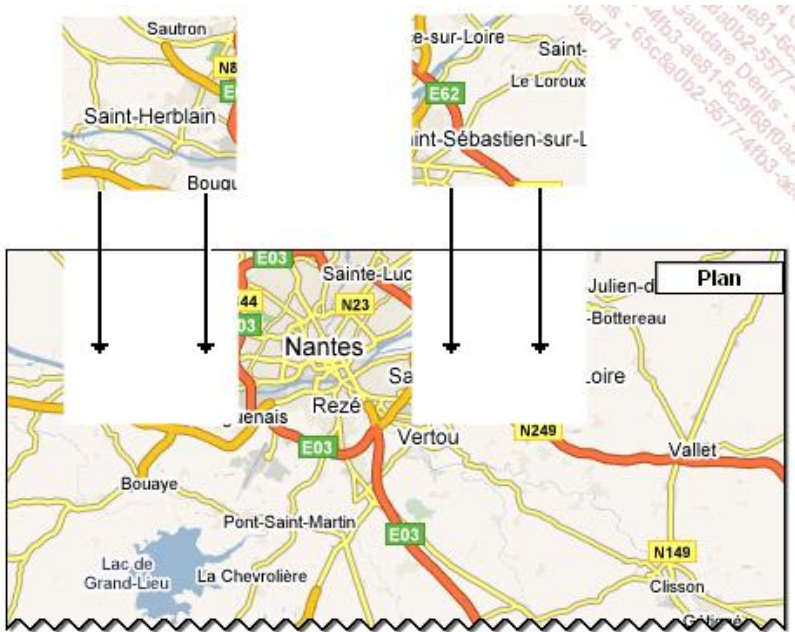
## 1. Google Maps

Est-il nécessaire de présenter Google Maps, ce service en ligne de cartes géographiques de calcul d'itinéraires et de vues satellites ?



En quoi le site Google Maps reprend-il l'approche AJAX ?

Il faut d'abord savoir qu'une carte est divisée en petits carrés. Pour les cartes classiques, ces petits morceaux au format PNG, ont une dimension de 256 x 256 pixels, et un nombre de couleurs limité à 256. Ce qui en fait des fichiers de taille réduite et ce qui implique un temps de transfert très faible.



Lorsque l'utilisateur déplace la carte dans un sens, par le simple glissement du curseur de la souris, cette action est prise en charge par le JavaScript. Celui-ci récupère la longitude, la latitude et l'intensité du zoom des carrés qui



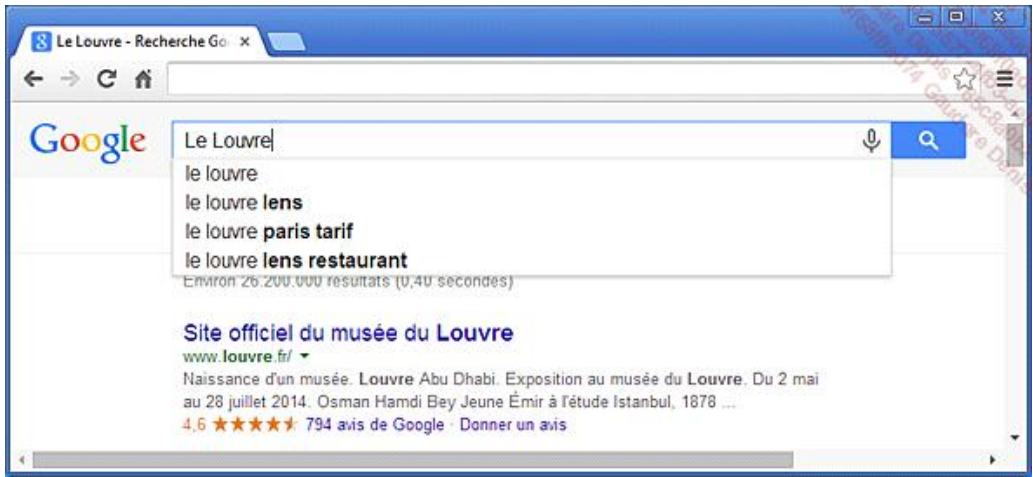
devront être dévoilés. L'objet XMLHttpRequest envoie une requête vers le serveur pour réclamer les petites images manquantes. À la réception, le DOM se chargera de les afficher dynamiquement à l'endroit adéquat, parfaitement alignées les unes avec les autres.

Toute cette opération se fait sans rechargement de la page, sans modification de l'adresse de la page, sans apparition de la barre d'avancement du téléchargement dans la barre d'état et sans réclamer l'installation d'un plug-in spécifique !

## 2. Google Suggest

Les suggestions automatiques lors de la saisie de mots-clés dans la barre de recherche de Google sont un autre exemple de l'utilisation d'AJAX.

À chaque lettre encodée, Google met à jour sa liste de suggestions de façon quasi instantanée et sans nécessiter de rafraîchissement de la page.



Le fonctionnement AJAX dans cette application peut être décomposé comme suit :

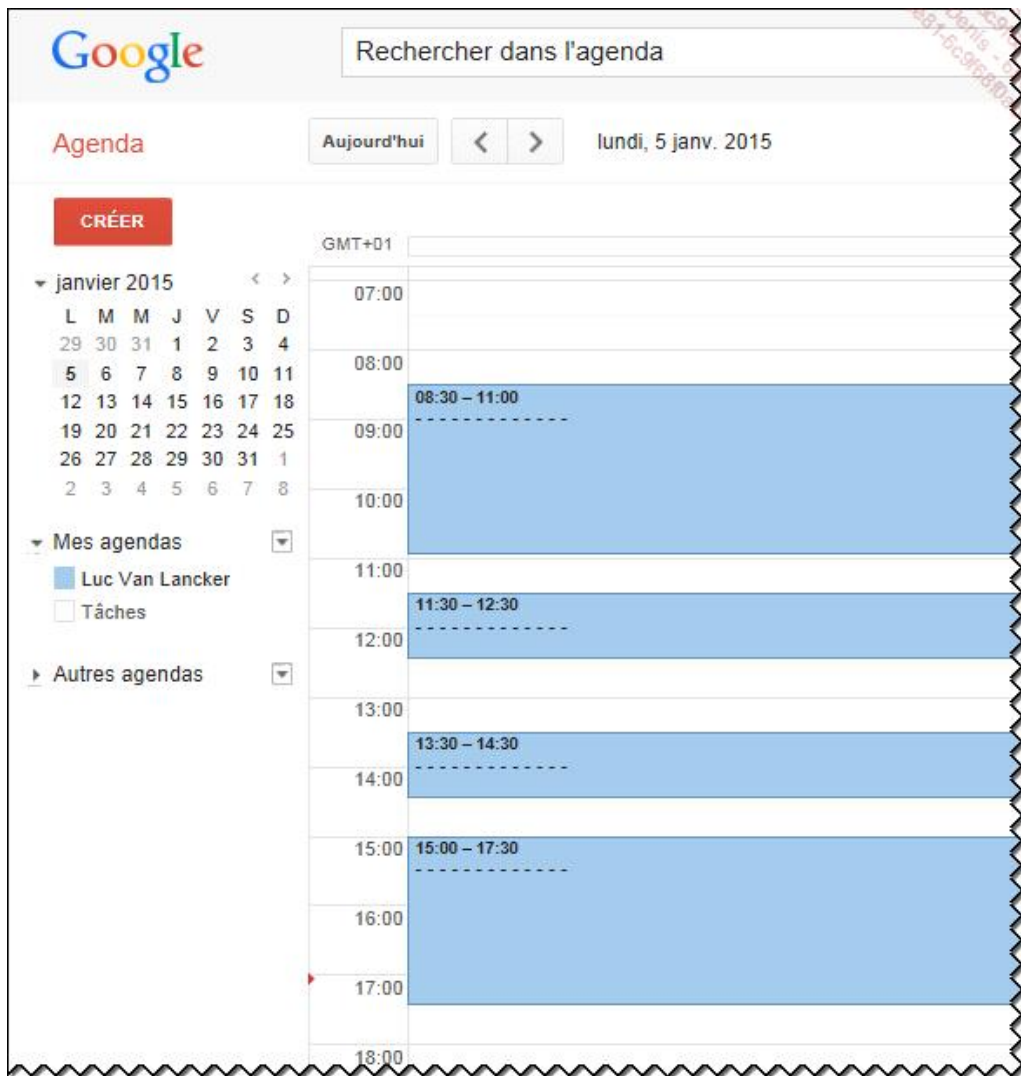
- Les lettres entrées dans la ligne de texte sont récupérées par du JavaScript.
- Une requête XMLHttpRequest est initiée avec comme paramètre la valeur de l'encodage.
- Celle-ci consulte une base de données située sur les serveurs de Google.
- Le serveur retourne une série de mots-clés qui pourraient répondre à la recherche effectuée.
- Le DOM crée une liste déroulante dans laquelle JavaScript se charge d'afficher les suggestions retournées par le serveur.

Ce système a été complété depuis son apparition. En effet, la base de données des suggestions tient compte de bien d'autres paramètres comme la géolocalisation de l'utilisateur ou de mots-clés déjà utilisés par celui-ci.

## 3. Google Agenda

Google Agenda, comme son nom l'indique, permet de gérer votre emploi du temps en toute simplicité, de façon très conviviale et colorée. Enregistrer des événements, modifier votre planning d'un simple cliquer/glisser de la souris, partager (avec votre permission) votre agenda, recevoir des notifications par e-mail, pop-up ou message sur votre téléphone portable, voici quelques-unes des fonctionnalités, très variées, de Google Agenda.

Le service en lui-même est accessible à partir de la page d'accueil de Google et nécessite l'ouverture (gratuite) d'un compte.



Le nombre important de fonctionnalités de Google Agenda illustre bien la richesse des applications AJAX : celles-ci permettent de mettre en place des sites qui n'ont plus rien à envier aux applications logicielles.

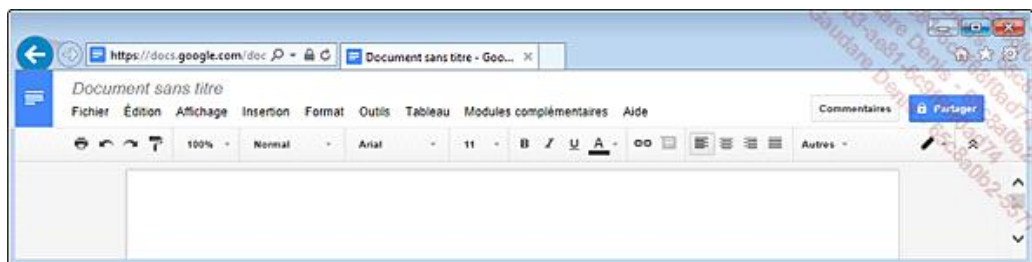
#### 4. Google Docs, Google Sheets et Google Slides

Créer, éditer, ou encore partager des documents demande toujours l'installation sur notre système d'un logiciel, en général Microsoft Office ou OpenOffice. Et si on pouvait faire tout cela avec seulement un navigateur web ?

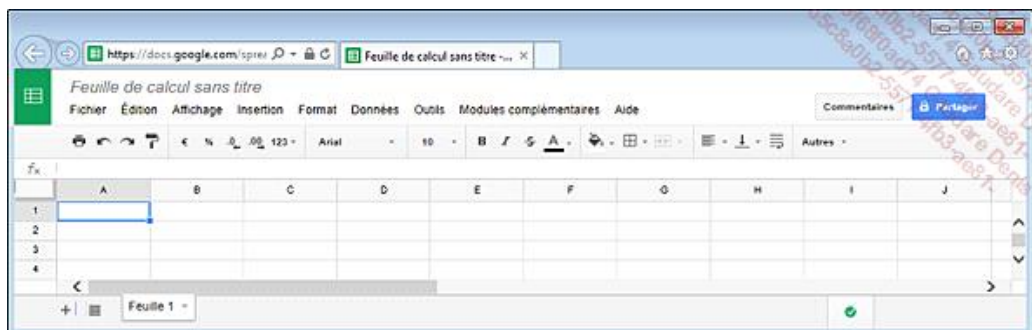
Google Docs propose une solution bureautique en ligne qui va vous permettre de disposer d'un traitement de texte, d'un tableur et d'un outil de présentation en ligne, sans aucune installation à effectuer. En outre ces documents pourront être conservés sur le cloud avec Google Drive.

Google Docs permet de créer des documents et de les partager pour faciliter le travail collaboratif. Le document peut être enregistré au format Word (.docx), OpenDocument (.odt), PDF (.pdf), texte brut (.txt), texte enrichi (.rtf) et page Web (.html).

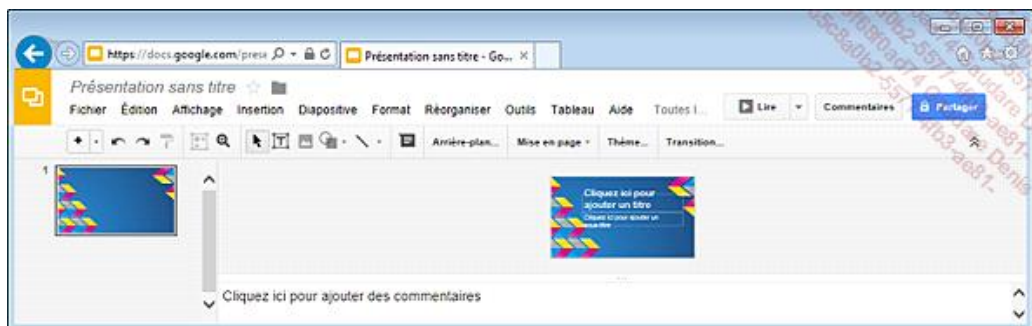




Google Sheets est le tableur de cette suite bureautique. La feuille de calcul peut être enregistrée au format Excel (.xlsx), OpenDocument (.ods) et PDF (.pdf).



Google Slides vous permet de réaliser des présentations de type PowerPoint. Elles peuvent être enregistrées au format Microsoft PowerPoint (.pptx), PDF (.pdf) ou dans un format image (.svg, .png ou .jpg).



# Avantages et désavantages

Il ne faut pas se laisser aveugler par l'effet AJAX. Tous les sites ne se prêtent pas à cette approche. Surtout que développer un site en AJAX sera plus difficile et plus long que les méthodes de publication utilisées à ce jour. Pour vous aider dans votre choix, voici quelques éléments qui vous aideront à prendre la bonne décision.

## 1. Avantages

AJAX, c'est principalement la révolution asynchrone ! La technique qui consiste à charger en arrière-plan des éléments de réponse aux interactions de l'utilisateur est au cœur du modèle AJAX. Il est certain que le potentiel est énorme et les développeurs ne se sont pas privés pour exprimer leur créativité. On remarque cependant que la technologie AJAX est plus adoptée chez nos amis anglo-saxons que dans les applications francophones.

L'objet XMLHttpRequest est également reconnu par les navigateurs des smartphones ou des tablettes. La technologie AJAX se révèle bien adaptée pour les applications mobiles car elle permet de ne charger qu'un contenu de taille réduite au lieu de devoir effectuer une requête complète pour recharger la page.

L'interface est plus réactive car seulement une petite partie est mise à jour, donnant ainsi l'impression à l'utilisateur que les changements se réalisent de façon instantanée.

Le temps d'attente de l'utilisateur est réduit car il n'est plus nécessaire de recharger toute la page pour obtenir une nouvelle information. Les applications AJAX apportent ainsi une fluidité inhabituelle aux pages web. Certains qualifient même AJAX comme étant le Web des impatientes. Je gage que certains lecteurs se reconnaîtront...

Le temps de transfert entre le client et le serveur est considérablement réduit car il suffit à ce dernier de transférer les éléments de la réponse au lieu d'envoyer toute la page avec des éléments le plus souvent redondants.

Un site AJAX bien réalisé sera agréable d'utilisation pour le visiteur. Ce qui ne peut que valoriser, en termes d'image, la communication avec l'utilisateur.

La maintenance du site sera plus aisée car, grâce à l'architecture mise en place par AJAX, il suffira de mettre à jour des fichiers de taille réduite, parfois simplement en format texte. Cette mise à jour pourra même être confiée à des personnes dont les compétences sont plus orientées bureautique que conception de sites web.

## 2. Désavantages

AJAX a beau utiliser des composants connus comme le JavaScript, le XML, le DOM et l'objet XMLHttpRequest, son code est pointu et délicat à mettre en place. Développer une application AJAX peut augmenter sensiblement le temps de conception et donc le coût. Il est admis de façon unanime qu'AJAX est plus difficile à mettre en place qu'une application web classique à cause de l'utilisation conjointe des différentes technologies. Cette affirmation est toutefois atténuée depuis l'apparition des frameworks JavaScript qui simplifient grandement l'encodage des requêtes AJAX.

AJAX comporte beaucoup de JavaScript. Les utilisateurs qui ont désactivé la prise en charge du JavaScript par leur navigateur ne pourront jamais accéder aux applications AJAX. On peut raisonnablement estimer que cette population est plus que réduite mais elle existe.

Le concept AJAX est orienté vers l'utilisateur. Cependant, par divers aspects, il risque de désorienter l'utilisateur moyen. Comme certaines parties de la page ne correspondront plus à la page initiale, les fonctions de l'historique et de la mise en favoris du navigateur risquent de ne pas être représentatives de l'état actuel de la page. De plus, cliquer sur le bouton "Précédent" du navigateur peut n'avoir aucun effet puisque l'adresse de la page est restée inchangée. L'API History du HTML5 remédie à ce problème et permet de revenir à la page précédente ou suivante.

L'accessibilité des applications AJAX avec les systèmes de lecture d'écran utilisés par les personnes visuellement déficientes est assez problématique. Déjà, la prise en compte du JavaScript y est très partielle et souvent aléatoire. En outre, vient s'ajouter l'objet XMLHttpRequest qui modifie, en cours de lecture, le contenu de la page. Ainsi, au moins dans un premier temps, il importera de prévoir une page alternative pour les personnes avec un handicap visuel.

L'objet XMLHttpRequest pouvant envoyer des requêtes du navigateur vers le serveur de façon transparente pour l'utilisateur, certaines personnes s'inquiètent quant au respect de la vie privée de l'internaute. On peut comprendre leur point de vue mais Internet est et restera un système ouvert dans toutes les implications du terme. Il ne nous appartient pas d'épiloguer ici sur cet éternel débat.

La technologie AJAX et l'objet XMLHttpRequest sont potentiellement vulnérables de par leur utilisation intensive du JavaScript et les requêtes fréquentes vers le serveur. Comme souvent lors de l'apparition d'une nouvelle technologie, des articles mettent en garde les développeurs sur l'utilisation de celle-ci. Sans vouloir mettre en doute la pertinence de ces chroniques, avec le recul de plusieurs années d'utilisation, il n'y a pas eu à notre connaissance de "catastrophe" sécuritaire imputée à l'utilisation d'AJAX.

Le référencement des pages en AJAX ne sera pas simple à réaliser. Comme pour les sites en JavaScript et les pages en Flash, le référencement réclamera toutes les attentions du concepteur. L'indexation par les moteurs de recherche n'est pas moins problématique car avec le système AJAX, une même page peut, selon les actions de l'internaute, avoir plusieurs contenus différents.

# Évolution ou révolution ?

S'il y a une question qui a alimenté, et qui alimentera encore, les forums de discussion, c'est bien celle-là. Tentons de faire le point au sujet des arguments développés.

## 1. Évolution

Il est évident qu'AJAX est une façon d'utiliser plusieurs technologies préexistantes.

Le JavaScript, omniprésent dans AJAX, fit ses premiers pas en 1996 avec le navigateur Netscape. Même normalisé par le ECMAScript, il restait cantonné à des tâches mineures. Avec le concept AJAX, il se montre enfin capable de gérer un processus complexe.

L'utilisation conjointe des feuilles de style CSS, du JavaScript et du DOM sur fond de page HTML n'est pas sans rappeler la définition du DHTML, qui lui aussi prônait déjà l'interactivité des pages web. L'objet XMLHttpRequest ne fait qu'apporter au DHTML la possibilité de récupérer des données du serveur.

Et l'objet XMLHttpRequest, parlons-en... Lui aussi n'est pas franchement nouveau car il était déjà été implanté dans Internet Explorer 5.0 en 1999 et passa tout ce temps inaperçu auprès des développeurs.

AJAX et le Web 2.0 ne seraient que le fruit de l'évolution naturelle qu'est la toile. Et des évolutions ou des mutations, le Web, implanté maintenant dans la vie quotidienne, en connaîtra encore.

## 2. Révolution

À la vue des exemples présentés lors du présent chapitre, il faut bien admettre que la mutation des pages web traditionnelles en véritables plates-formes applicatives va modifier, de façon significative, les usages de la toile. Avec la richesse des interfaces, avec la multiplication des fonctionnalités, avec les avancées de l'ergonomie et de la facilité d'utilisation, c'est assurément une nouvelle façon de naviguer qui s'instaure.

La véritable nouveauté d'AJAX ne réside pas tant dans son aspect technologique, mais bien dans la nouvelle stratégie qu'il souhaite mettre en place. C'est la première fois depuis l'apparition du Web que l'utilisateur, ou autrement dit le consommateur, est placé au centre des préoccupations des développeurs. AJAX permet la conception de sites réactifs, ergonomiques, intuitifs, axés sur le consommateur du Web et ses nouvelles attentes d'utilisation du Web.

Avec le recul de plusieurs années, la révolution AJAX réside assurément dans l'importance prise par cette technologie dans la conception des applications web. En témoignent les nombreux sites ayant adopté AJAX, les nombreux tutoriels qui lui sont consacrés, ainsi que la place qu'il occupe dans les frameworks JavaScript.

En outre AJAX trouve de nouvelles perspectives dans la multiplication des applications pour smartphones ou tablettes. Pour celles-ci, en permettant de faire évoluer la page par le chargement de fichiers de taille réduite et de faire l'impasse sur le rechargement complet de la page, il apporte une fluidité appréciable aux applications qui l'ont adopté.

# Les pré-requis


L'objectif de cet ouvrage, après avoir parcouru les éléments nécessaires, est de faire percevoir au lecteur le fonctionnement concret et les énormes potentialités des applications AJAX.

Force est de constater, après la lecture du chapitre Présentation générale d'AJAX, que, du HTML en passant par les feuilles de style CSS, le JavaScript, le XML, le DOM et l'objet XMLHttpRequest, les différentes technologies ne manquent pas.

Afin d'éviter de proposer une encyclopédie de la conception de pages web, il semble judicieux de limiter les matières abordées dans ce livre. À ce stade de l'intérêt du lecteur pour les nouvelles perspectives de la publication sur le Net, il paraît raisonnable de penser que celui-ci possède déjà des notions basiques ou avancées du langage HTML ainsi que des feuilles de style CSS.

AJAX met en œuvre, à travers différents composants comme le DOM et l'objet XMLHttpRequest, un code pointu et a priori complexe qui réclamera toute votre attention. Une certaine aisance dans le langage HTML et les feuilles de style CSS ne peut que faciliter l'étude du lecteur.

Pour parfaire vos connaissances sur le HTML5 et les feuilles de style CSS, vous pouvez vous reporter à l'ouvrage "HTML5 et CSS3 - Maîtrisez les standards des applications Web" aux Éditions ENI - Ressources Informatiques du même auteur.

 La connaissance de langage HTML et des feuilles de style CSS est un pré-requis indispensable à une étude fructueuse de cet ouvrage consacré à AJAX.

Tous les exemples du livre sont écrits en HTML5.

# Les limites de ce livre

La lecture du chapitre Présentation générale d’AJAX a fait découvrir que l’approche AJAX permettait de traiter dans le navigateur de l’utilisateur (application côté client) des fichiers externes à la page, présents sur le serveur (application côté serveur).

Les données de ces fichiers externes peuvent être traitées par des langages de programmation, comme par exemple le PHP ou l’ASP. Ce qui élargit encore, et de façon considérable, les compétences nécessaires pour produire des pages web en AJAX. À ce niveau, la frontière entre le lecteur averti et le professionnel nous semble franchie.

Ainsi, nous limiterons l’étude de cet ouvrage sur AJAX à des applications côté client qui récupéreront des fichiers externes, au format XML ou texte, et qui les afficheront de façon asynchrone dans la page web.



## 1. Un navigateur récent

### Google Chrome

Toujours à la pointe des technologies de la publication sur le Web, que ce soit pour les feuilles de style CSS ou les API JavaScript du HTML5, le navigateur Google Chrome est sans conteste le navigateur préféré des développeurs. Les versions se succèdent à un rythme très (trop ?) élevé. Par son système de mise à jour automatique, l'utilisateur est assuré de toujours disposer de la dernière version. C'est avec plaisir que l'on voit sa part de marché augmenter de mois en mois.

### Firefox

Le navigateur de la communauté Mozilla reste un excellent navigateur même s'il semble être quelque peu en perte de vitesse. Depuis qu'il a adopté lui aussi le système de mise à jour automatique, vous serez également assurés d'utiliser la dernière mouture de ce navigateur.

### Internet Explorer

Avec les versions 10 et 11 d'Internet Explorer, Microsoft a partiellement rattrapé son retard vis-à-vis de Google Chrome et Firefox, en particulier au niveau des feuilles de style CSS3. À l'opposé de ses concurrents, Microsoft préfère éditer de nouvelles versions de son navigateur de façon ponctuelle. Si à chacune de leur apparition Internet Explorer reprend un coup de jeune, au fil des mois le navigateur reprend à chaque fois du retard par rapport à la concurrence.

## 2. Un éditeur de texte

Pour la suite de notre étude, n'importe quel éditeur de texte brut sans aucune mise en forme fera l'affaire. Sous Windows, vous pouvez utiliser le Bloc-notes ou Notepad. Le pendant du bloc-notes pour les Macintosh est TextEdit. Citons en outre des éditeurs de texte en version gratuite comme Notepad++ pour Windows et TextWrangler pour Mac.

# Les outils côté serveur

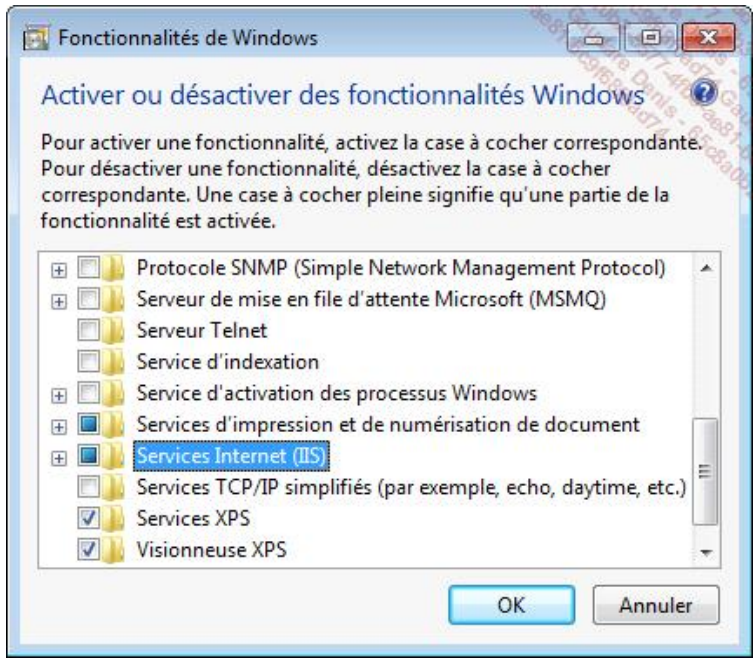
Si pour l'étude du JavaScript, du XML et du DOM, soit respectivement les chapitres Le JavaScript, Introduction au XML et Introduction au XSL, un éditeur de texte et votre navigateur suffiront largement, la suite du livre consacrée à l'objet XMLHttpRequest et à AJAX (chapitres Le DOM (Document Object Model), L'objet XMLHttpRequest, L'approche AJAX et AJAX par l'exemple) nécessite la mise en place d'un serveur web local, aussi appelé serveur web personnel ou virtuel.

Ce serveur local vous permettra de tester le code sans avoir à vous connecter à un serveur distant et à y télécharger vos fichiers par FTP pour pouvoir les tester.

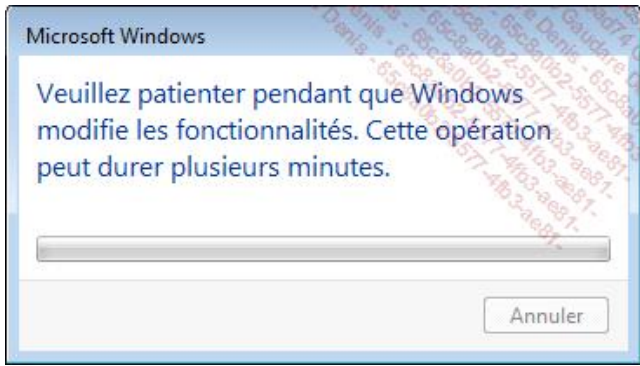
## 1. Le serveur local Microsoft IIS

Le logiciel serveur web IIS (*Internet Information Services*) est présent dans l'installation de Windows. Cependant vous devez l'activer :

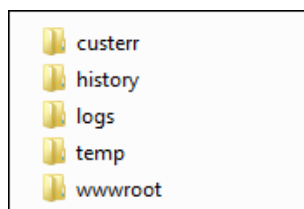
- **Démarrer - Panneau de configuration - Désinstaller un programme**
- Dans la fenêtre qui s'ouvre alors, choisissez **Activer ou désactiver des fonctionnalités de Windows**.



- Cliquez sur **OK** pour démarrer l'installation.

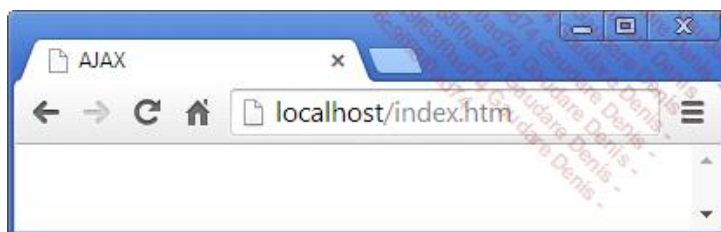


Vous trouverez alors sur le disque dur C un nouveau dossier inetpub contenant à son tour différents sous-dossiers.



Les fichiers du serveur local doivent être déposés ou enregistrés dans le sous-dossier inetpub/wwwroot.

Pour accéder à un fichier, par exemple index.htm, on saisira « localhost/index.htm » dans la barre d'adresse du navigateur.



## 2. Le serveur local EasyPHP

EasyPHP est une plate-forme de développement web comprenant le renommé serveur Apache ainsi qu'un interpréteur de scripts PHP, le serveur de bases de données MySQL et l'outil d'administration PhpMyAdmin. Initialement prévu pour le développement de scripts PHP en local, son serveur Apache vous permet également de tester en local vos applications AJAX.

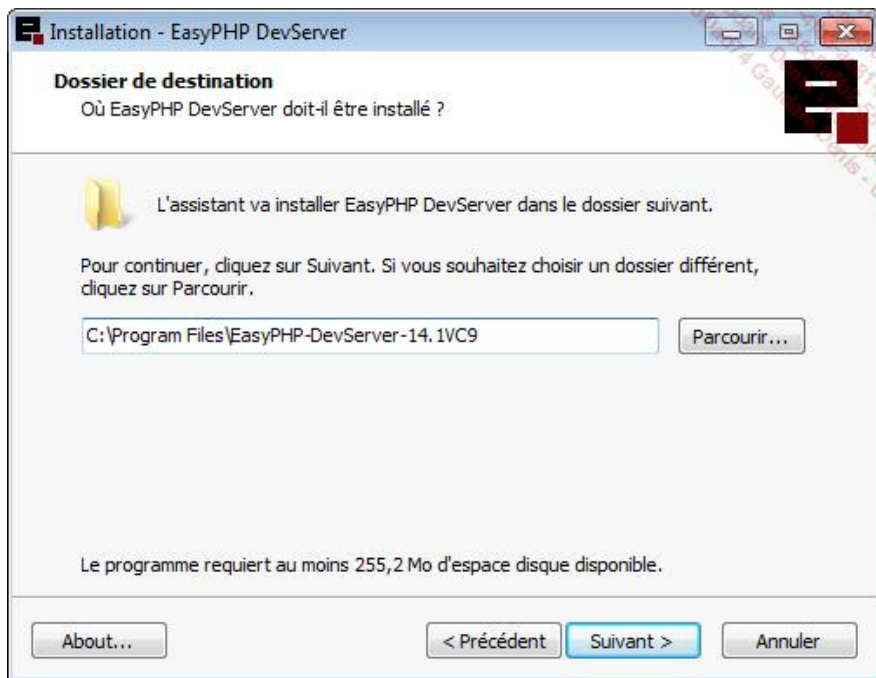


EasyPHP est téléchargeable à l'adresse : [www.easyphp.org/download.php](http://www.easyphp.org/download.php). Précisons que EasyPHP est un outil qui a largement fait ses preuves et qu'il est entièrement gratuit.

Après avoir sélectionné la langue qui sera utilisée par l'assistant d'installation, celle-ci débute.



La suite de programmes sera accessible sur le disque dur **C - Programmes - EasyPHP**.



L'installation est terminée.

Les fichiers à tester devront être situés dans le sous-dossier **Programmes - EasyPHP - data - localWeb**.

On saisit ensuite l'URL du fichier dans la barre d'adresse, par exemple `index.php`, `localhost/index.php` ou `127.0.0.1/index.php`.

