


AJAX

Introduction


- AJAX = Asynchronous JavaScript and XML
- AJAX n'est pas un nouveau langage de programmation, mais une manière d'utiliser des standards existants
- AJAX permet d'échanger des données avec un serveur et mettre à jour une page web sans recharger toute la page

Exemple : Gmail

Gmail [Calendar](#) [Documents](#) [Photos](#) [Groups](#) [Web](#) [more ▾](#) tester@gmail.com | [Settings](#) | [Older version](#) | [Help](#) | [Sign out](#)

 Search Mail Search the Web [Show search options](#) [Create a filter](#)

[Compose Mail](#)

[Inbox \(5\)](#)
[Starred](#) ★
[Chats](#) 
[Sent Mail](#)
[Drafts \(1\)](#)
[All Mail](#)
[Spam](#)
[Trash](#)

[Contacts](#)

▶ Bob T. Monkey
 Search, add, or invite

▼ **Labels**
[friends \(3\)](#)
[mailing](#)
[To Do](#)
[vacation](#)
[work \(2\)](#)
[Edit labels](#)

▶ Invite a friend

[Archive](#) [Report Spam](#) [Delete](#) [More Actions ▾](#) [Refresh](#) 1 - 16 of 16

Select: [All](#), [None](#), [Read](#), [Unread](#), [Starred](#), [Unstarred](#)

| | | | | |
|--------------------------|------------------------------|----------|---|--------|
| <input type="checkbox"/> | Caitlin, me (2) | work | Flight number? - Can you let me know which flight | Apr 19 |
| <input type="checkbox"/> | me, Nathan (3), Draft | friends | Gift ideas for Caitlin - Any ideas | Apr 19 |
| <input type="checkbox"/> | Paige, me (3) | | Shopping trip - Awesome, I'll see you then. We can grab a | Apr 18 |
| <input type="checkbox"/> | Nicola Brennan | | Friday drinks? - You guys free on Friday? We're thinking i | Apr 18 |
| <input type="checkbox"/> | ★ Lizzie, me (2) | | Africa airfares - Cool, I'll check it out. Sounds like a good i | Apr 18 |
| <input type="checkbox"/> | ★ Caitlin Roran | work | Conference budget - We'll need to sit down with I | Apr 18 |
| <input type="checkbox"/> | ★ Nathan Woodward | friends | Birthday cake - Any idea which kind of cake is | Apr 18 |
| <input type="checkbox"/> | Lizzie, me (2) | friends | Cancelling tennis on Friday - No problem, it was | Apr 18 |
| <input type="checkbox"/> | ★ Caitlin Roran | work | March expense reports - I've attached the recent r | Apr 17 |
| <input type="checkbox"/> | me, Caitlin (4) | work | Meeting with Mitchell - I think it would be best if we | Apr 17 |
| <input type="checkbox"/> | me, Nicola (2) | | Photos - Can you send me that photo you mentioned earl | Apr 17 |
| <input type="checkbox"/> | ★ Paige Stevens | friends | Camera shopping - Did you get the link for the i | Apr 17 |
| <input type="checkbox"/> | Zach, me (2) | work | Timesheets? - I think we should ask Wayne for his | Apr 17 |
| <input type="checkbox"/> | Nathan Woodward | | Surprise party - I was wondering if maybe you and Susan | Apr 17 |
| <input type="checkbox"/> | Nicola Brennan | vacation | Fishing - Hi guys, Mark and I have arranged th | Apr 17 |
| <input type="checkbox"/> | Lizzie Astley | friends | Trip to Africa - Hey, so do you still want to go? It | Apr 17 |

Select: [All](#), [None](#), [Read](#), [Unread](#), [Starred](#), [Unstarred](#)

[Archive](#) [Report Spam](#) [Delete](#) [More Actions ▾](#) [Refresh](#) 1 - 16 of 16

Tasks
 Add task e.g. TPS report 5pm
[Set task view here ▾](#)

▼ **Today**
● Pick up the milk
Today

▼ **Tomorrow**
● Finish TPS reports
Wed @ 5:00pm

▼ **This Week**
● Return library books
Friday

▼ **Next Week**
● Pay electricity bill
Dec 27

▼ **Anytime**
● Call Caitlin
● Take over the world
● Apply for new passport
● Order Heroes DVD
● Prepare presentation
● Get bananas
● Research Africa airfares

[Options ▾](#) [Refresh](#)

Exemple : Facebook



The image is a screenshot of a web browser displaying Mark Zuckerberg's Facebook profile. The browser's address bar shows the URL `www.facebook.com/zuck?sk=wall`. The Facebook interface includes a top navigation bar with the Facebook logo, a search bar, and icons for home, messages, and notifications. On the left side, there is a large profile picture of Mark Zuckerberg and a sidebar with links to 'Wall', 'Info', 'Share Profile', and 'Report/Block This Person'. The main content area on the right displays the name 'Mark Zuckerberg' followed by his work and education details: 'Works at Facebook', 'Studied Computer Science at Harvard University', 'Lives in Palo Alto, California', 'Knows English, Mandarin Chinese', 'From Dobbs Ferry, New York', and 'Born on May 14, 1984'. Below this is a section titled 'Wall' with a sub-header 'RECENT ACTIVITY'. The first activity shows a green speech bubble icon and the text 'I like dangerous thoughts.' on Samuel W. Lessin's status. The second activity is a post by Mark Zuckerberg with a small profile picture, the text 'Steve, you've done so much good for the world already. I hope you get better soon.', and a timestamp 'January 17 at 11:43am via iPhone'. Below the post, a blue bar indicates '150 people like this.' with a thumbs-up icon.

← → ↻ 🏠 🌐 `www.facebook.com/zuck?sk=wall`

facebook 👤 💬 🔔

Search 🔍

Mark Zuckerberg

🏢 Works at Facebook 🎓 Studied Computer Science at Harvard University 🏠 Lives in Palo Alto, California 🗣️ Knows English, Mandarin Chinese 🏡 From Dobbs Ferry, New York 📅 Born on May 14, 1984

Wall

RECENT ACTIVITY

💬 "I like dangerous thoughts." on Samuel W. Lessin's status.

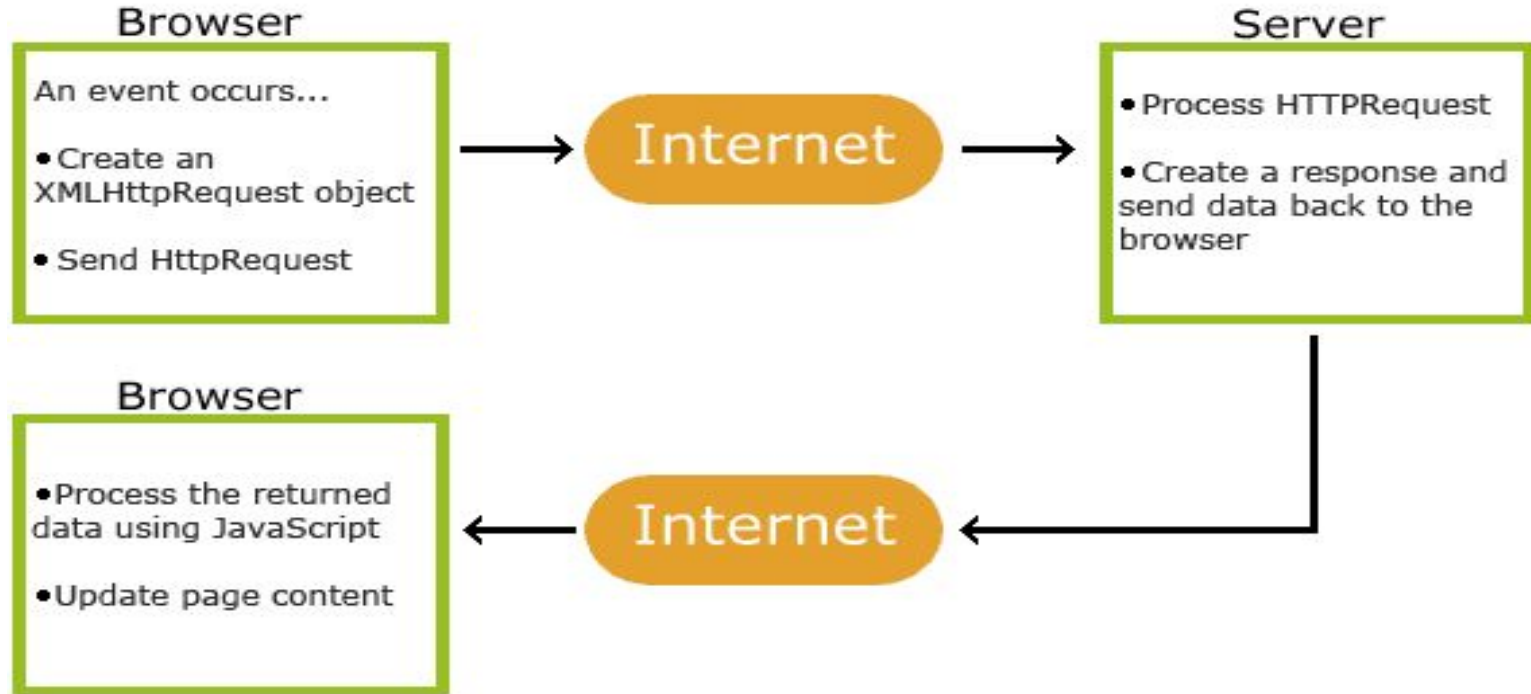
 **Mark Zuckerberg**
Steve, you've done so much good for the world already. I hope you get better soon.
📱 January 17 at 11:43am via iPhone

👍 150 people like this.

Standards utilisés

- HTML/CSS
- JavaScript/DOM
- XML (ou JSON)

Fonctionnement



L'objet XMLHttpRequest

- L'approche AJAX repose sur l'objet XMLHttpRequest
- Cet objet est défini par le world wide web consortium (w3.org)
- Cet objet permet:
 - d'envoyer une requête HTTP vers un serveur
 - de récupérer les données renvoyées par le serveur
 - ... et ce sans recharger complètement la page web
- Créer un objet XMLHttpRequest (IE7+, Firefox, Chrome,...)

```
xhr = new XMLHttpRequest();
```

L'objet XMLHttpRequest

- L'objet XMLHttpRequest est supporté par IE depuis la version 7
- L'objet originel est un contrôle ActiveX appelé XMLHTTP développé par Microsoft dans IE 5.5

```
axo = new ActiveXObject("Microsoft.XMLHTTP");
```


L'objet XMLHttpRequest

- Méthode pour garantir la rétro-compatibilité

```
function getXMLHttpRequest(){  
    var xhr;  
    if (window.XMLHttpRequest)  
    { //code pour IE7+, Firefox, Chrome, Safari, Opera  
        xhr = new XMLHttpRequest();  
    }  
    else  
    { //code pour IE5 et IE6  
        xhr = new ActiveXObject("Microsoft.XMLHTTP");  
    }  
    return xhr;  
}
```

Envoyer une requête au serveur

- Pour envoyer une requête, deux méthodes sont utilisées

La première spécifie le type de requête, l'url ainsi que le fait si la requête est asynchrone :

```
xhr.open("method", "url", async);
```

- method : le type de requête GET ou POST
- url : l'adresse du fichier/script à interroger
- async: true (asynchrone) ou false (synchrone)

La seconde envoie la requête au serveur :

```
xhr.send("string");
```

Exemples d'envois de requête

- Get

```
xhr.open("GET","demo_get2.asp?fname=Henry&lname=Ford",true);  
xhr.send();
```
- Post

```
xhr.open("POST","ajax_test.asp",true);  
  
xhr.setRequestHeader("Content-type",  
    "application/x-www-form-urlencoded");  
  
xhr.send("fname=Henry&lname=Ford");
```

Synchrone ou Asynchrone

- Synchrone: l'exécution du javascript se stoppe jusqu'à la réception de la réponse de l'objet XMLHttpRequest
- Asynchrone: du code javascript peut être exécuté durant l'attente de la réponse de l'objet XMLHttpRequest

Réponse du serveur

- `responseText()`

Méthode de `xmlHttpRequest` permettant de traiter la réponse du serveur comme une chaîne de caractères

```
document.getElementById("myDiv").innerHTML = xhr.responseText;
```

- `responseXML()`

Méthode de `xmlHttpRequest` permettant de traiter la réponse du serveur comme un fichier XML

Exemple pour responseXML()

```
<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
</CATALOG>
```

Exemple pour responseXML()

```
var xdoc = xhr.responseXML;
var x = xdoc.getElementsByTagName("TITLE");

var txt = "";
for (i = 0; i < x.length; i++)
{
    txt = x[i].nodeValue + "<br />";
}

document.getElementById("myDiv").innerHTML = txt;
```

État d'XMLHttpRequest

- La propriété `readyState` permet de connaître l'état de la requête:
 - 0 : L'objet XHR a été créé, mais pas encore initialisé (la méthode `open` n'a pas encore été appelée)
 - 1 : L'objet XHR a été créé, mais pas encore envoyé (avec la méthode `send`)
 - 2 : La méthode `send` vient d'être appelée
 - 3 : Le serveur traite les informations et a commencé à renvoyer des données
 - 4 : Le serveur a fini son travail, et toutes les données sont réceptionnées
- L'événement « `readystatechange` » permet de capturer les changements d'état de la propriété `readyState`

État d'XMLHttpRequest

- La propriété status permet de connaître le code d'état de la requête:
 - 200: Requête traitée avec succès
 - 400: La syntaxe de la requête est erronée
 - 401: Une authentification est nécessaire pour accéder à la ressource
 - 403: L'authentification est refusée.
Contrairement à l'erreur 401, aucune demande d'authentification ne sera faite
 - 404: Ressource non trouvée
 - 500: Erreur interne du serveur

Exemple AJAX complet

```
var xhr = new XMLHttpRequest();

xhr.onreadystatechange = function()
{
    if (xhr.readyState == 4
        && (xhr.status == 200 || xhr.status == 0))
    {
        alert("OK");
    }
};

xhr.open("GET", "handlingData.php?a=2", true);
xhr.send(null);
```

La fonction de Callback

- Permet de passer une fonction en argument

```
function funcA(callback)
{
    // instruction...
    // instruction...
    // instruction...
    callback();
}

function funcB()
{
    alert("Plop");
}

funcA(funcB);
```

Exemple AJAX + Callback

```
function request(callback) {  
    var xhr = new XMLHttpRequest();  
  
    xhr.onreadystatechange = function() {  
        if (xhr.readyState == 4 && (xhr.status == 200 || xhr.status == 0)) {  
            callback(xhr.responseText);  
        }  
    };  
  
    xhr.open("GET", "handlingData.php", true);  
    xhr.send(null);  
}  
  
function readData(sData) {  
    // On peut maintenant traiter les données sans encombrer l'objet XHR.  
    if (sData == "OK") {  
        alert("Ok!");  
    }  
    else {  
        alert("Ko!");  
    }  
}  
  
request(readData);
```

Merci pour votre attention.

