

Asynchronous JavaScript and XML (Ajax)

Jean-Claude Charr

Maître de conférences

IUT de Belfort – Montbéliard

Université de Franche Comté



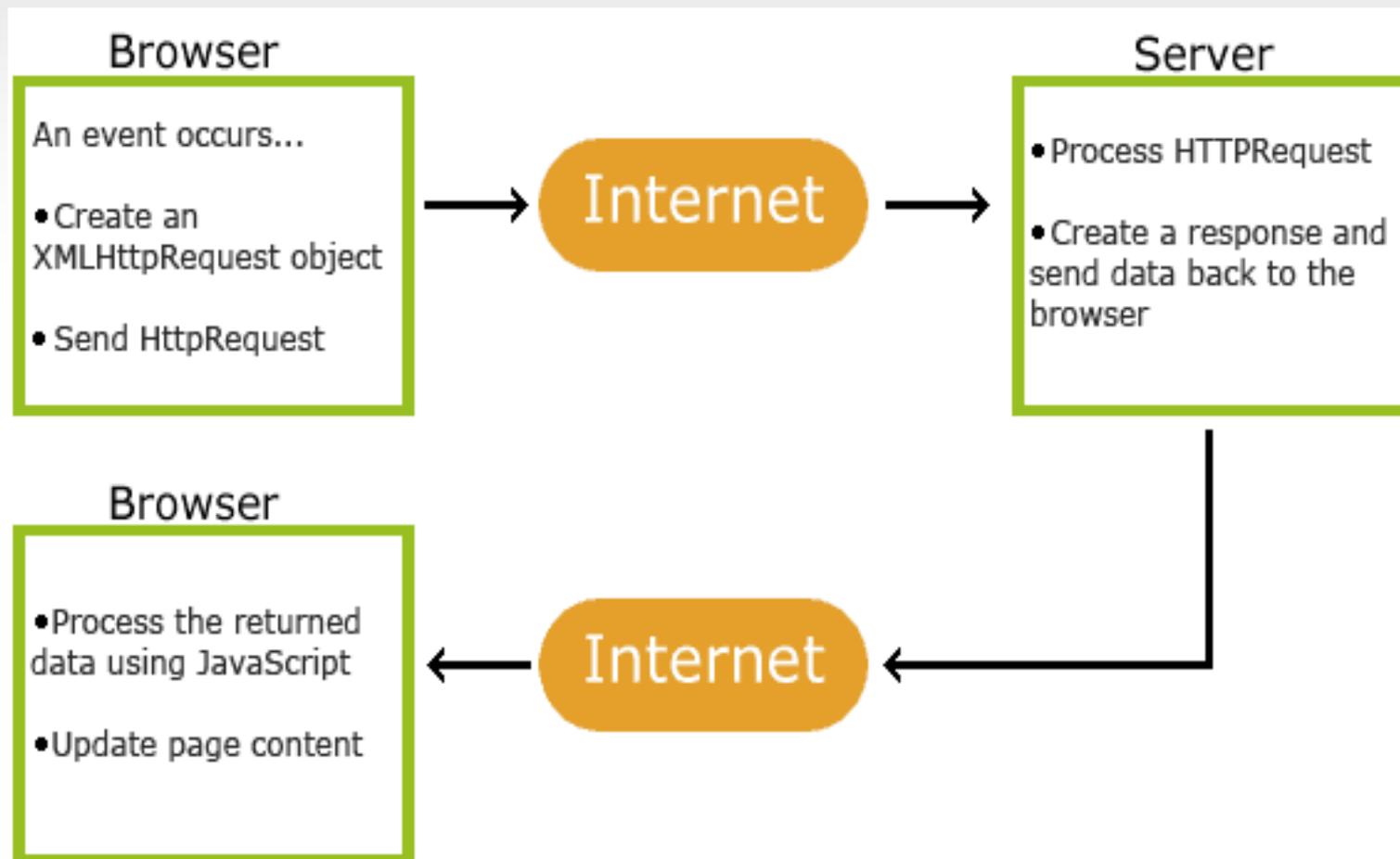
Description générale

- AJAX est une technique utilisée pour créer des pages web dynamiques.
- Elle permet la mise à jour des pages webs, sans devoir les recharger, en échangeant d'une manière asynchrone des données avec le serveur web
- Exemples : Google Maps, youtube, Facebook, ...

Comment Ajax fonctionne (1/2)

- AJAX est basé sur les standards de l'Internet :
- **Objet XMLHttpRequest** : pour échanger des données d'une manière asynchrone avec le serveur
- **JavaScript/DOM** : pour afficher et interagir avec les informations
- **CSS** : pour mettre en forme les données
- **JSON** : format utilisé pour les données transférées

Comment Ajax fonctionne (2/2)



Objet XMLHttpRequest

- Instancier un objet XMLHttpRequest :

```
xmlhttp=new XMLHttpRequest();
```

- Configurer la requête avec la méthode open() et l'envoyer au serveur avec la méthode send() de l'objet XMLHttpRequest :

- xmlhttp .open(méthode,url,async) : configurer la requête

méthode : POST ou GET

url : l'adresse de la page web appelée

async : true pour asynchrone ou false pour synchrone

- Xmlhttp.send(data) : envoyer la requête au serveur

data : préciser les données à envoyer avec la méthode POST

Les méthodes GET et POST (1/2)

- La méthode GET est plus rapide et facile à utiliser pour l'envoi des requêtes
- Il faut utiliser la méthode POST :
 - quand c'est interdit l'utilisation de fichiers caches
Ex : pendant la mise à jour d'un fichier sur le serveur
 - pour envoyer un grand fichier de données (POST n'a pas de limitation sur la taille des données)
 - pour envoyer des données contenant des caractères spéciaux.
 - Pour envoyer les données confidentielles

Les méthodes GET et POST (2/2)

- Exemple de requête "GET" avec Envoi de données :

```
xmlhttp.open("GET","/api/client?fname=joe&lname=Ford",true);  
xmlhttp.send();
```

- Envoyer des données avec la requête "POST" :

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

Synchrone ou asynchrone?

- Ajax => **Asynchrone**

```
xmlhttp.onreadystatechange=function(){  
    if (xmlhttp.readyState==4 && xmlhttp.status==200)  
        document.write(xmlhttp.responseText); }  
xmlhttp.open("GET","/api/clients",true);  
xmlhttp.send();
```

- **Synchrone** pour les petites requêtes

```
xmlhttp.open("GET","/api/clients",false);  
xmlhttp.send();  
document.write(xmlhttp.responseText);
```

onreadystatechange (1/2)

Cet évènement est déclenché lorsque :

1. La connexion avec le serveur est établie
 2. La requête est reçue par le serveur
 3. La requête est traitée
 4. Le traitement est terminé et la réponse est prête
- La propriété **onreadystatechange** contient la fonction a exécuté après un changement de l'état de la requête
 - La propriété **readyState** contient l'état de la requête : 0, ..., 4
 - La propriété **status** contient l'état de la réponse : 200 ou 404

onreadystatechange (2/2)

Exemple :

```
xmlhttp.onreadystatechange=function(){  
    if (xmlhttp.readyState==4 && xmlhttp.status==200){  
        document.getElementById("myDiv").innerHTML=  
        xmlhttp.responseText;  
    }  
}
```

Exemple Ajax avec Flask (1/4)

Page HTML :

```
<html>
  <head>
    <script type="text/javascript" src="script.js"/>
  </head>
  <body>
    <h3>Start typing a name in the input field below:</h3>
    First name: <input type="text" id="txt1" onkeyup=
    "showHint(this.value)" />
    <p>Suggestions: <span id="txtHint"></span></p>
  </body>
</html>
```

Exemple Ajax avec Flask (2/4)

Fichier JavaScript :

```
function showHint(data){
    xmlhttp=new XMLHttpRequest();
    xmlhttp.onreadystatechange=function(){
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
            document.getElementById("txtHint").innerHTML=
                xmlhttp.responseText;
    }
    xmlhttp.open("GET", "/api/hint?data="+data, true);
    xmlhttp.send();
}
```

Exemple Ajax avec Flask (3/4)

```
from flask import request
@app.route('/api/hint', methods=['GET'])
def getHint():
    mydb = mysql.connector.connect(host="localhost", port="8889",
    user="root", password="root", database="flask")

    data=request.args["data"]

    cursor = mydb.cursor()

    sql="SELECT Name FROM user WHERE Name like '"+data+"%"
    cursor.execute(sql)
```

Exemple Ajax avec Flask (4/4)

```
results = cursor.fetchall()
```

```
cursor.close()
```

```
response = ""
```

```
for result in results:
```

```
    response +=result['Name']+ ", "
```

```
return response
```

JSON

- JSON pour JavaScript Object Notation
- C'est un format pour stocker les données dans une chaîne de texte
- JSON est un format **indépendant** de tout langage mais il utilise la syntaxe de JavaScript pour la description des objets
- En JavaScript, JSON peut être converti en objet JS
- Les fichiers JSON ont une extension .json

Syntaxe

- JSON a une syntaxe similaire aux objets JavaScript :
- Les données ont le format : "nom" : "valeur"
- Les données sont séparées par des ,
- Les objets sont entourés d'accolades
- Les tableaux sont entourés de crochets
- Une valeur peut être un nombre, une chaîne de caractères, un booléen, un tableau ou un objet

Ex : { "employees": [
 {"fName":"John" , "lName":"Doe"},
 {"fName":"Anna" , "lName":"Smith"}
] }

Ajax avec JSON (1/3)

Fichier HTML :

```
<html>
  <head>
    <script type="text/javascript" src="script.js"/>
  </head>
  <body>
    <div id="txtCDInfo">
      <button onclick="getCD()">Get CD info</button>
    </div>
  </body>
</html>
```

Ajax avec JSON (2/3)

La réponse du serveur en JSON :

```
{ "CATALOG": [ { "CD":  
                { "TITLE": "Empire Burlesque",  
                  "ARTIST": "Bob Dylan"      }  
              },  
              { "CD":  
                { "TITLE": "Hide your heart",  
                  "ARTIST": "Bonnie Tyler"   }  
              }  
            ]  
}
```

Ajax avec JSON (3/3)

Fichier JavaScript :

```
function getCD(){
    xmlhttp=new XMLHttpRequest();
    xmlhttp.onreadystatechange=function(){
        if (xmlhttp.readyState==4 && xmlhttp.status==200){

            var x=JSON.parse(xmlhttp.responseText);
            txt="";
            for (i=0;i<x.CATALOG.length;i++)
                txt=txt+x.CATALOG[i].CD.TITLE+", ";
            document.getElementById('txtCDInfo').innerHTML=txt;
        }
        xmlhttp.open("GET", "/api/cds", true);
        xmlhttp.send();
    }
}
```