

INF157 - Utilisation des Réseaux

Licence 3 Informatique

Arnaud Pecher (repris par Damien Magoni)

Bureau 322, Bâtiment A30, LaBRI
Université de Bordeaux

Licence 3 Informatique - Bordeaux

- 1 Présentation du fonctionnement AJAX
- 2 DOM
- 3 TP : tutoriel de Siddh
- 4 En guise de conclusion

Bibliographie

- D. Cabasson, *Ajax - une autocomplétion pas à pas*
- G. Donat, *Introduction à AJAX et interaction avec PHP*
- B. W. Perry, *Ajax à 200*, O'Reilly
- N. Pied, *Ajax : vos premiers pas dans les nouvelles technologies*
- Siddh, *Web 2.0, allez plus loin avec AJAX et XMLHttpRequest*

- 1 Présentation du fonctionnement AJAX
- 2 DOM
- 3 TP : tutoriel de Siddh
- 4 En guise de conclusion

Comparaison

- **navigation web "classique"** : navigation de page en page, initiées par l'utilisateur (clics sur hyperliens, soumission de formulaires) ;
- **AJAX** : échanges de données via HTTP entre le navigateur et le serveur, **transparents** pour l'utilisateur, et **rafraichissement de parties** de la page uniquement.

Avantages

- interface utilisateur plus conviviale et réactive ;
- moins de données échangées.

Web 2.0

Utilisation d'AJAX pour de nouveaux usages (clients mail évolués, bureautique en ligne, blogs etc).

AJAX - technologies employées

- **A**synchrone : envoi de requêtes HTTP, sans attente de la réponse ;
- **J**avascript : initiées par le navigateur ;
- **X**ML : contenant des données formatées en XML.

Fondements

- l'objet JavaScript **XmlHttpRequest** pour la gestion des échanges HTTP ;
- le modèle objet des navigateurs de représentation d'une page web : **Document Object Model** (DOM) pour la gestion d'une partie de la page uniquement ;

JavaScript

Langage de programmation **interprété par le navigateur**.
Le code Javascript peut être inséré directement dans la section d'en-tête du document ou dans le corps du document :

```
<SCRIPT type='text/javascript'>...</SCRIPT>
```

Des **fonctions** peuvent être **associées à différents évènements** (interaction de l'utilisateur, modifications de la page).

DOM

Document Object Model est une **interface de programmation** (API) **pour des documents**. Cette interface ressemble étroitement à la structure des documents (XHTML) qu'elle modélise.

Méthodes

- **open(méthode, url, mode) : ouvre une connexion avec le serveur**
 - **méthode** : méthode HTTP qui sera envoyée (GET ou POST) ;
 - **url** : url du serveur ;
 - **mode** : asynchrone (true) ou synchrone (false).
- **setRequestHeader(nom, val) : affecte la valeur val à l'entête HTTP nom qui sera envoyé ;**
- **send(params) : fixe le corps de la requête HTTP ;**
- **abort() : abandon de la requête ;**

Variables

- **onreadystatechange** : fonction appelée en cas de changement de la variable **readyState** ;
- **readyState** : état de l'objet - valeurs possibles :
 - 0 : non initialisé ;
 - 1 : requête HTTP prête ;
 - 2 : requête HTTP envoyée ;
 - 3 : réception de la réponse HTTP en cours ;
 - 4 : réception de la réponse HTTP terminée.
- **status** : code HTTP de la réponse ;
- **statusText** : message associé au code HTTP ;
- **responseText** : corps de la réponse ;
- **responseXML** : corps de la réponse sous forme de document XML.

Instanciation

L'instanciation de l'objet XMLHttpRequest nécessite un code spécifique pour Internet Explorer :

```
<script type='text/JavaScript'>  
    .  
    .           function getXhr(){  
    .           var xhr = null;.  
    .           if(window.XMLHttpRequest) // Firefox et autres  
    .           xhr = new XMLHttpRequest();  
    .           .  
    .           else if(window.ActiveXObject){ // Internet Explorer.  
    .           try {  
    .           .           xhr = new ActiveXObject("Msxml2.XMLHTTP");  
    .           .           } catch (e) {  
    .           .           xhr = new ActiveXObject("Microsoft.XMLHTTP");  
    .           .           }  
    .           .  
    .           }  
    .           else { // XMLHttpRequest non supporté par le navigateur.  
    .           .           alert("Votre navigateur ne supporte pas les objets XMLHttpRequest...");  
    .           .           xhr = false;.  
    .           .  
    .           }.  
    .           return xhr  
    .           }  
    .  
</script>
```

Exemple

Le code suivant instancie un objet XMLHttpRequest lors de l'appui du bouton et affiche ce qu'a retourné le script PHP `ajaxTest1` dans une boîte de notification.

Code client

```
<html><head>
.
.
  <title>Tutoriel Ajax</title>
  <script type='text/JavaScript' src='getXhr.js'></script>
  <script type='text/JavaScript'>
.
.
    /** Méthode qui sera appelée sur click du bouton */
    function go(){
.
.
        var xhr = getXhr();
.
.
        xhr.onreadystatechange = function(){
.
.
            // si code ok et tout reçu
            if(xhr.readyState == 4 && xhr.status == 200){alert(xhr.responseText);}
.
.
        }
.
.
        xhr.open("GET", "./ajaxTest1.php", true);
        xhr.send(null);
.
.
    }
  </script>
</head><body><input type='button' value='Allo ??' onclick='go()' /></body></html>
```

Code php

```
<?php
echo "Oui, Ajax, je suis là!";
?>
```

A vous de jouer

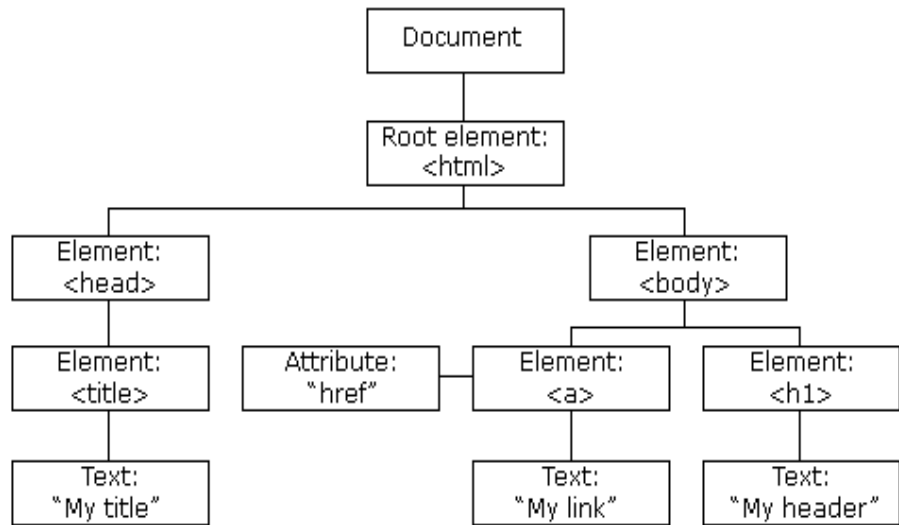
- 1 tester le code précédent en récupérant les 3 fichiers : `getXhr.js`, `ajaxEtatsXmlHttpRequest.html` et `ajaxTest.php` ;
- 2 modifier ce code pour afficher dans des boites de notification les états intermédiaires de l'objet `XmlHttpRequest` ;
- 3 utiliser un analyseur de trame (comme `wireshark`) pour intercepter le trafic HTTP généré par cette application

Solution

```
<html><head>
  <title>Tutoriel Ajax</title>
  <script type='text/JavaScript' src='getXhr.js'></script>
  <script type='text/JavaScript'>
    /** Méthode qui sera appelée sur click du bouton */
    function go(){
      var xhr = getXhr();
      xhr.onreadystatechange = function(){
        if (xhr.readyState == 0) alert("Non initialisé");
        if (xhr.readyState == 1) alert("Requête HTTP Get envoyée");
        if (xhr.readyState == 2) alert("Données envoyées");
        if (xhr.readyState == 3) alert("En cours de réception");
        if (xhr.readyState == 4) alert("Réception terminée");
        // si code ok et tout reçu
        if(xhr.readyState == 4 && xhr.status == 200){alert(xhr.responseText);}
      }
      xhr.open("GET", "./ajaxTest1.php", true);
      xhr.send(null);
    }
  </script>
</head><body><input type='button' value='Allo ??' onclick='go()' /></body></html>
```

- 1 Présentation du fonctionnement AJAX
- 2 DOM**
- 3 TP : tutoriel de Siddh
- 4 En guise de conclusion

Arbre DOM



Variables

- **nodeName** : nom du sommet ;
- **nodeValue** : valeur du sommet ;
- **nodeType** : type du sommet.

Element type	NodeType
Element	1
Attribute	2
Text	3
Comment	8
Document	9

Sélection de sommet(s)

Sélection directe

- `document.getElementById(String id) ;` : sélection des sommets par leur identifiant ;
- `document.getElementsByTagName(String tag) ;` : sélection des sommets par leur mot-clef ;

Exemple

```
var x=document.getElementsByTagName("p");
for (var i=0;i<x.length;i++)
{
  // do something with each paragraph
}
var y=document.getElementById('maindiv').getElementsByTagName("p");
```

x : tous les paragraphes.

y : tous les paragraphes fils des éléments d'identifiant "maindiv".

Principales méthodes d'un sommet

- `appendChild()` : ajout d'un fils au sommet courant ;
- `createElement()` : création d'un élément à insérer dans l'arbre ;
- `createTextNode()` : création d'une feuille contenant du texte ;
- `getAttribute()` : obtenir la valeur du sommet courant ;
- `insertBefore()` : insertion élément

Principales variables d'un sommet

- **childNodes** : liste des fils du sommet courant ;
- **firstChild** : premier fils ;
- **id** : identifiant ;
- **lastChild** : dernier fils ;
- **nodeName** : nom du sommet courant ;
- **parentNode** : sommet ;
- **style** : style associé au sommet ;

Document

L'objet Document représente l'ensemble du document et peut être utilisé pour accéder à n'importe quel élément.

Variables de type collection

Collection	Description	IE	F	O	W3C
anchors[]	Returns a reference to all Anchor objects in the document	4	1	9	Yes
forms[]	Returns a reference to all Form objects in the document	4	1	9	Yes
images[]	Returns a reference to all Image objects in the document	4	1	9	Yes
links[]	Returns a reference to all Area and Link objects in the document	4	1	9	Yes

Document : variables

Property	Description	IE	F	O	W3C
body	Gives direct access to the <body> element				
<u>cookie</u>	Sets or returns all cookies associated with the current document	4	1	9	Yes
<u>domain</u>	Returns the domain name for the current document	4	1	9	Yes
<u>lastModified</u>	Returns the date and time a document was last modified	4	1	No	No
<u>referrer</u>	Returns the URL of the document that loaded the current document	4	1	9	Yes
<u>title</u>	Returns the title of the current document	4	1	9	Yes
<u>URL</u>	Returns the URL of the current document	4	1	9	Yes

Document : méthodes

Method	Description	IE	F	O	W3C
<u>close()</u>	Closes an output stream opened with the document.open() method, and displays the collected data	4	1	9	Yes
<u>getElementById()</u>	Returns a reference to the first object with the specified id	5	1	9	Yes
<u>getElementsByName()</u>	Returns a collection of objects with the specified name	5	1	9	Yes
<u>getElementsByTagName()</u>	Returns a collection of objects with the specified tagname	5	1	9	Yes
<u>open()</u>	Opens a stream to collect the output from any document.write() or document.writeln() methods	4	1	9	Yes
<u>write()</u>	Writes HTML expressions or JavaScript code to a document	4	1	9	Yes
<u>writeln()</u>	Identical to the write() method, with the addition of writing a new line character after each expression	4	1	9	Yes

Exercice

Afficher un texte quand rien de spécial ne survient ; le drapeau français quand la souris est au-dessus du lien "français", et le drapeau anglais quand la souris est au-dessus du lien "english".

URL de démonstration :

<http://membres.lycos.fr/forumstock/tuto/exemple1.html>

Source :

<http://www.siteduzero.com/tuto-3-3558-1-xhtml-javascript-et-dom.html>

XHTML

```
<body>
<p id="testbox">If you see an image appearing in the current box when
  your mouse is over the link below javascript is enabled on your
  computer</p>
<p class="javascript"><a href="test.html" onmouseover="javascript:flag
  (1)" onmouseout="javascript:flag(0)">English</a> <a href="test.html"
  onmouseover="javascript:flag(2)" onmouseout="javascript:flag(0)">
  Francais</a></p>
</body>
```

JavaScript

```
function flag(type){
  if( type == 1 ){//si la variable vaut 1
    //Suppression de l'ancien contenu
    var paragraphe = document.getElementById("testbox");
    var old_contenu = paragraphe.firstChild;
    paragraphe.removeChild(old_contenu);
    //Ajout du nouveau contenu
    var img = document.createElement("img");
    img.setAttribute("src", "english.jpg");
    img.setAttribute("alt", "english flag");
    paragraphe.appendChild(img);
```

Exercice

Faire un formulaire affichant une description du champ sur lequel on clique dans un paragraphe sur la page. URL de démonstration :

`http://membres.lycos.fr/forumstock/tuto/exemple2.html`

Source :

`http://www.siteduzero.com/tuto-3-3558-1-xhtml-javascript-et-dom.html`

XHTML

```
<p id="testbox">If you see information appearing in the current box  
  when you click in a text field javascript is enabled on your  
  computer</p>  
<div class="javascript">  
<form action="test.html" method="post">  
<fieldset>  
<label>Nom <input type="text" name="nom" onFocus="javascript:infos(1)"  
  onBlur="javascript:infos(4)" /></label><br />
```

JavaScript

```
function infos(number){  
  if(number == 1){  
    //Suppression de l'ancien contenu  
    var paragraphe = document.getElementById("testbox");  
    var old_contenu = paragraphe.firstChild;  
    paragraphe.removeChild(old_contenu);  
    //Ajout de l'information  
    var info = document.createTextNode("Entrez votre nom de  
      famille.");  
    paragraphe.appendChild(info);  
  } else if(number == 4) ...
```

- 1 Présentation du fonctionnement AJAX
- 2 DOM
- 3 TP : tutoriel de Siddh**
- 4 En guise de conclusion

A vous de jouer : listes déroulantes liées

But

Créer dans un même formulaire deux listes déroulantes liées **Auteurs** et **Livres** : la liste déroulante **Livres** contient la liste de l'auteur sélectionné **sans actualisation de la page**.

Correspondance auteurs-livres : commandes sql

```
CREATE TABLE `auteur` (  
  `id` tinyint(4) NOT NULL auto_increment,  
  `nom` varchar(50) NOT NULL,  
  PRIMARY KEY (`id`)  
);  
  
insert into `auteur` values  
(1,'Clive Cussler'), (2,'Harlan Coben'), (3,'Franck Herbert'), (4,'Pierre Bordages');  
  
CREATE TABLE `livre` (  
  `id` tinyint(4) NOT NULL auto_increment,  
  `titre` varchar(50) NOT NULL,  
  `idAuteur` tinyint(4) default NULL,  
  PRIMARY KEY (`id`)  
);  
  
insert into `livre` values  
(1,'Odyssee',1), (2,'Sahara',1), (3,'Dragon',1), (4,'Une chance de trop',2), (5,'Ne le dis a personne',2),  
(6,'Disparu à jamais',2), (7,'Dune',3), (8,'La barriere de santaroga',3), (9,'Les guerriers du silence',4),  
(10,'La citadelle hyponeros',4), (11,'Terra mater',4);
```

Quelques indications

- 1 l'objet `XmlHttpRequest` utilise la méthode `POST` pour communiquer le choix de l'auteur au script PHP `ajaxTPLivres.php` ;
- 2 ceci implique de préciser l'encodage : `xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded')` ;
- 3 pour la liste déroulante `livre`, l'englober dans une balise `div` utilisant le style `display : inline` ;
- 4 ajuster la liste des livres avec `document.getElementById('livre').innerHTML = listeLivres` où `listeLivres` contient la réponse du script `ajaxTPLivres.php`.

Liste des livres (serveur)

```
<?php
echo "<select name='livre'>";
if(isset($_POST["idAuteur"])){
    mysql_connect("localhost", "root", "");
    mysql_select_db("M2BDI");
    $res = mysql_query("SELECT id,titre FROM livre
        WHERE idAuteur=".$_POST["idAuteur"]." ORDER BY titre");
    while($row = mysql_fetch_assoc($res)){
        echo "<option value='". $row["id"]."'>". $row["titre"]."</option>";
    }
}
echo "</select>";
?>
```

Liste des auteurs (client)

```
<html><head>
  <title>Tutoriel Ajax (XHTML + JavaScript + XML)</title>
  <script type='text/JavaScript' src='getXhr.js'></script>
  <script type='text/JavaScript'>
    function go(){
      var xhr = getXhr();
      xhr.onreadystatechange = function(){
        if(xhr.readyState == 4 && xhr.status == 200){
          livres = xhr.responseText;
          // innerHTML pour rajouter les options a la liste
          document.getElementById('livre').innerHTML = livres;
        }
      }
      xhr.open("POST","ajaxTPLivres.php",true);
      xhr.setRequestHeader('Content-Type','application/x-www-form-urlencoded');
      // poster les arguments: l'id de l'auteur
      sel = document.getElementById('auteur');
      idauteur = sel.options[sel.selectedIndex].value;
      xhr.send("idAuteur="+idauteur);
    }
  </script>
</head>
<body>
  <form>
    <fieldset style="width: 500px">
      <legend>Liste liées</legend>
      <label>Auteurs</label>
      <select name='auteur' id='auteur' onchange='go()'>
        <option value='-1'>Aucun</option>
        <?
          mysql_connect("localhost","root","");
          mysql_select_db("M2BDI");
          $res = mysql_query("SELECT * FROM auteur ORDER BY nom");
          while($row = mysql_fetch_assoc($res)){
            echo "<option value='".$row["id"]."'">".$row["nom"]."</option>";
          }
        </select>
      <label>Livres</label>
      <div id='livre' style='display:inline'>
      <select name='livre'>
        <option value='-1'>Choisir un auteur</option>
      </select>
      </div>
    </fieldset>
  </form>
</body>
</html>
```

- 1 Présentation du fonctionnement AJAX
- 2 DOM
- 3 TP : tutoriel de Siddh
- 4 En guise de conclusion**

Trame d'un service web / en AJAX

Ethernet

Données

- **Ethernet** : adressage réseau local
- IP : adressage Internet
- TCP : transport fiable pour Internet
- HTTP : transport des données pour le Web
 - du contenu d'une page (en XHTML), générées en PHP ou autre ;
 - des feuilles de style (en CSS) ;
 - ou d'autres documents (WSDL, XML (AJAX) ...)
- XML : transport des données structurées (AJAX) ;
- SOAP : adapté à l'invocation du service (Service Web).

Trame d'un service web / en AJAX

Ethernet

IP

Données

- **Ethernet** : adressage réseau local
- **IP** : adressage Internet
- **TCP** : transport fiable pour Internet
- **HTTP** : transport des données pour le Web
 - du contenu d'une page (en XHTML), générées en PHP ou autre ;
 - des feuilles de style (en CSS) ;
 - ou d'autres documents (WSDL, XML (AJAX) ...)
- **XML** : transport des données structurées (AJAX) ;
- **SOAP** : adapté à l'invocation du service (Service Web).

Trame d'un service web / en AJAX

Ethernet

IP

TCP

Données

- **Ethernet** : adressage réseau local
- **IP** : adressage Internet
- **TCP** : transport fiable pour Internet
- **HTTP** : transport des données pour le Web
 - du contenu d'une page (en XHTML), générées en PHP ou autre ;
 - des feuilles de style (en CSS) ;
 - ou d'autres documents (WSDL, XML (AJAX) ...)
- **XML** : transport des données structurées (AJAX) ;
- **SOAP** : adapté à l'invocation du service (Service Web).

Trame d'un service web / en AJAX

Ethernet

IP

TCP

HTTP

Données

- **Ethernet** : adressage réseau local
- **IP** : adressage Internet
- **TCP** : transport fiable pour Internet
- **HTTP** : transport des données pour le Web
 - du contenu d'une page (en XHTML), générées en PHP ou autre ;
 - des feuilles de style (en CSS) ;
 - ou d'autres documents (WSDL, XML (AJAX) ...)
- **XML** : transport des données structurées (AJAX) ;
- **SOAP** : adapté à l'invocation du service (Service Web).

Trame d'un service web / en AJAX

Ethernet

IP

TCP

HTTP

XML

Données

- **Ethernet** : adressage réseau local
- **IP** : adressage Internet
- **TCP** : transport fiable pour Internet
- **HTTP** : transport des données pour le Web
 - du contenu d'une page (en XHTML), générées en PHP ou autre ;
 - des feuilles de style (en CSS) ;
 - ou d'autres documents (WSDL, XML (AJAX) ...)
- **XML** : transport des données structurées (AJAX) ;
- **SOAP** : adapté à l'invocation du service (Service Web).

Trame d'un service web / en AJAX

Ethernet

IP

TCP

HTTP

XML

SOAP

Données

- **Ethernet** : adressage réseau local
- **IP** : adressage Internet
- **TCP** : transport fiable pour Internet
- **HTTP** : transport des données pour le Web
 - du contenu d'une page (en XHTML), générées en PHP ou autre ;
 - des feuilles de style (en CSS) ;
 - ou d'autres documents (WSDL, XML (AJAX) ...)
- **XML** : transport des données structurées (AJAX) ;
- **SOAP** : adapté à l'invocation du service (Service Web).