

AJAX

[MCours.com](http://MCours.com)

# Qu'est ce qu'AJAX ?

---

- ◉ **Asynchronous JavaScript And XML.**
- ◉ Ajax a exploité les technologies XML et java script
- ◉ AJAX est un concept qui permet de faire des appels asynchrones au serveur depuis le client.
- ◉ Lors de ces appels, le serveur retournera du XML qui sera "récupéré" par javascript et traité.
- ◉ Nous verrons que nous pouvons tout aussi bien faire transiter du texte et faire des appels synchrones si l'on veut.

# Problématique

---

- Avant toute chose, il serait bon de faire un point sur le processus classique de consultation d'un site ou d'une application web :
  - Vous saisissez une adresse dans votre navigateur.
  - Cette "requête" finie par arriver sur le serveur web qui héberge la page en question.
  - Le serveur vous retourne du texte au format HTML ou XHTML et éventuellement des images, feuilles de style, fichiers JavaScript, applets java ....
  - Votre navigateur les interprète et vous affiche la page.
  - Vous êtes déconnecté du serveur web.
- Donc, quand vous cliquez sur un lien, vous recommencez ce processus en entier avec une nouvelle page.
  - Dans le cas où un formulaire se trouve sur la page, vous envoyez les données sur le serveur qui vous répondra après traitement de ces données.

# Apport de ajax

---

- L'utilisation d'AJAX va chambouler un peu cette organisation car à tout moment vous pouvez aller chercher des informations sur le serveur pour :
  - Ajouter des éléments a la page
  - Modifier le contenu d'un "bout de la page"
  - Insérer des données dans une base.
  - ...

# Utilisation de Ajax

---

## ○ L'objet **XmlHttpRequest**

- AJAX se base sur l'utilisation d'un composant embarqué dans presque tous les navigateurs récents.
- Par contre, vous vous doutez bien que le comportement va varier en fonction de ces derniers.
- Pour pouvoir utiliser AJAX, il nous faut donc créer en javascript un objet que l'on nomme **XmlHttpRequest** ou **xhr** pour les intimes,
- comme son nom l'indique, cet objet nous permet de faire des requêtes http pour échanger du XML.

# Création de l'objet XMLHttpRequest

---

```
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;
}
```

## Propriétés et méthode de l'objet xhr

---

### ⦿ **open("méthode", "url", flag):**

- Ouvrir la connexion avec le serveur.
- **méthode** -> "GET" ou "POST"
- **url** -> l'url à laquelle on va envoyer notre requête.
  - Si la méthode est GET, on met les paramètres dans l'url.
  - **flag** -> true si l'on veut un dialogue asynchrone, sinon, false

## Propriétés et méthode de l'objet xhr

---

- **setRequestHeader("nom", "valeur"):**
  - Assigne une valeur à un header HTTP qui sera envoyé lors de la requête.
  - Par exemple, pour un POST :
    - nom -> "Content-Type«
    - valeur -> "application/x-www-form-urlencoded"

# Propriétés et méthode de l'objet xhr

---

- **send("params"):**
  - Envoi la requête au serveur.
  - Si la méthode est GET, on met null en paramètre.
  - Si la méthode est POST, on met les paramètres a envoyer, sous la forme :
    - "nomparam1=valeur1&nomparam2=valeur2".
- **abort()** : Abandonne la requête.
- **onreadystatechange** :
  - Ici, on va lui affecter notre fonction java script qui sera exécutée à chaque "changement d'état" de notre objet.

# Propriétés et méthode de l'objet xhr

---

## ○ **readyState :**

- C'est cette propriété qu'on va tester dans le `onreadystatechange`.
- Elle représente l'état de l'objet et peut prendre plusieurs valeurs :
  - 0 -> Non initialisé.
  - 1 -> Ouverture (`open()` vient de s'exécuter).
  - 2 -> Envoyé (`send()` vient de s'exécuter).
  - 3 -> En cours (des données sont en train d'arriver).
  - 4 -> Prêt (toutes les données sont chargées).

# Propriétés et méthode de l'objet xhr

---

- **status :**

- **Le code de la réponse du serveur.**
  - 200 -> OK.
  - 404 -> Page non trouvée.
  - ...

- **statusText :** Le message associé à status.

- **responseText :** La réponse retournée par le serveur, au format texte.

- **responseXML:** La réponse retournée par le serveur, au format dom XML.

# Premier exemple

---

- script.jsp :
- Ce script retourne client le texte bonjour ajax suivi d'un nombre aléatoire compris entre 1 et 100

```
<%
```

```
    out.println((int)(Math.random()*100));
```

```
%>
```

# Premier exemple : `index.htm`

```
<html>
<head>
<title>Tutoriel Ajax (XHTML + JavaScript + XML)</title>
<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("Le navigateur ne supporte pas les objets XMLHttpRequest...");
        xhr = false;    }
    return xhr
}
```

## Premier exemple : `index.htm` (suite)

```
/**      * Méthode qui sera appelée sur le click du bouton */
function go(){
    var xhr = getXhr()
    // On définit ce qu'on va faire quand on aura la réponse
    xhr.onreadystatechange = function(){
        // si on a tout reçu et que le serveur est ok
        if(xhr.readyState == 4 && xhr.status == 200){
            document.getElementById('rep').innerHTML = xhr.responseText;
        }
        xhr.open("GET", "script.jsp", true);
        xhr.send(null);
    }
}
</script>
</head>
<body>
    <input type='button' value='Test Ajax' onclick='go()' />
    <div id="rep"> Ici la réponse ajax </div>
</body>
</html>
```

# Application

On considère Une base de données MySQL qui contient deux tables « auteurs » et « livres ». Un auteur peut créer plusieurs livres. La structure de cette base de données est la suivante :

id	nom
1	Clive Cussler
2	Harlan Coben
3	Franck Herbert
4	Pierre Bordages

id	titre	idAuteur
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

- On souhaite créer une application web utilisant la technologie Ajax qui permet de :
  - Afficher tous les auteurs dans un tableau HTML
  - En cliquant sur un auteur, afficher tous les livres de cet auteur.

# Structure de la base

---

- `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);`

# Structure de la base

## ○ Table auteur

id	nom
1	Clive Cussler
2	Harlan Coben
3	Franck Herbert
4	Pierre Bordages

## □ Table livre

id	titre	idAuteur
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

# Vue à réaliser

Liste liées

Auteurs

Aucun



Livres

Choisir un auteur



Choisir un auteur

Liste liées

Auteurs

Aucun



Livres

Choisir un auteur



Aucun

Clive Cussler

Franck Herbert

Pierre Bordages

Robert Ludlum

# auteurs .php

```
<?
$conn=mysql_connect("localhost","root","") or die(mysql_error());
mysql_select_db("db_aj",$conn)or die(mysql_error());
$req="select * from auteur";
$rsAut=mysql_query($req)or die(mysql_error());
?>
<html><body><form>
Auteur: <select name="idAuteur" id="idAuteur" onChange="go()">
    <option value="-1">Aucun</option>
    <? while($aut=mysql_fetch_assoc($rsAut)){ ?>
<option value="<? echo $aut['id']?>"><? echo $aut['nom']?></option>
    <? } ?>
</select>
<div id="livre" style="display:inline">
Livres:<select name="idLivre" id="idLivre">
    <option value="-1">Choisir un auteur</option>
</select>
</div>
</form>
</body>
```

## auteurs .php (suite)

```
<head>
<title>Listes liées Ajax</title>
<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("Le navigateur ne supporte pas les objets
XMLHttpRequest...");
        xhr = false;
    }
    return xhr;
}
```

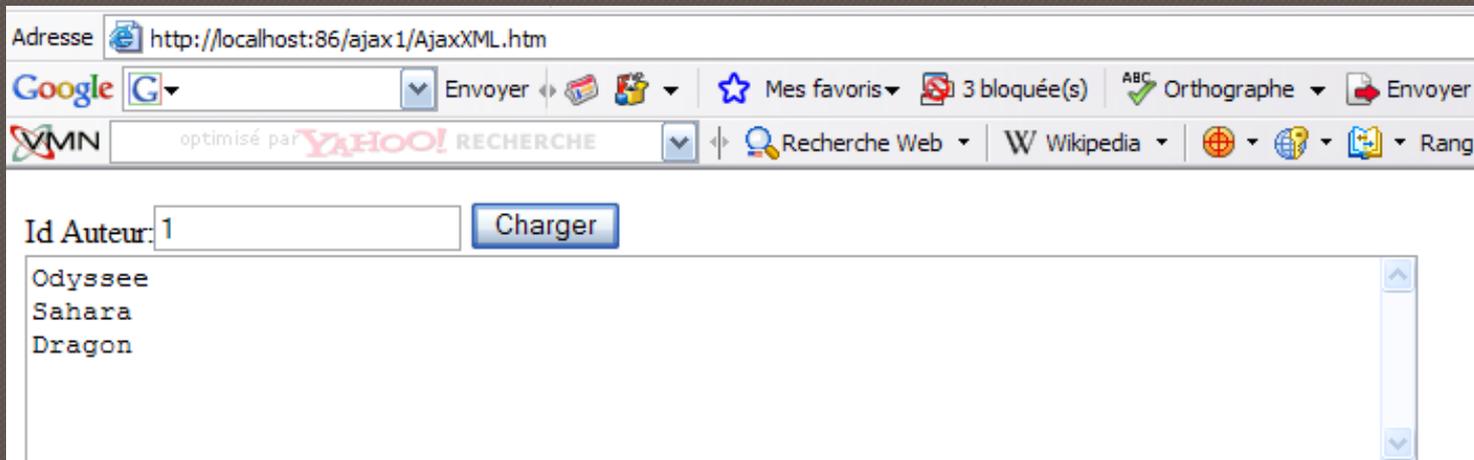
# auteurs .php (suite)

```
function go(){
var xhr=getXhr();
// On définit ce qu'on va faire quand on aura la réponse
xhr.onreadystatechange=function(){
// si on a tout reçu et que le serveur est ok
if((xhr.readyState==4)&&(xhr.status==200)){
lesLivres=xhr.responseText;
// On se sert de innerHTML pour rajouter les options a la liste
document.getElementById('livre').innerHTML=lesLivres;
}
// Ici on va voir comment faire une requête post
xhr.open("POST","ajaxLivres.php",true);
xhr.setRequestHeader('Content-Type','application/x-www-form-urlencoded');
// on récupère l'objet liste des auteurs
sel=document.getElementById("idAuteur");
idAuteur=sel.options[sel.selectedIndex].value;
xhr.send("idauteur="+idAuteur);
}
</script>
</head></html>
```

```
<?
$idauteur=$_POST['idauteur'];
$conn=mysql_connect("localhost","root","") or die(mysql_error());
mysql_select_db("db_aj",$conn)or die(mysql_error());
$req="select * from livre where idAuteur=$idauteur";
$rs=mysql_query($req)or die(mysql_error());
?>
<select name="livre">
<option value="-1">selectionner un livre</option>
<? while ($liv=mysql_fetch_assoc($rs)){?>
<option value="<? echo($liv['id'])?>"><? echo($liv['titre'])?>
</option>
<? }?>
</select>
```

# Ajax et XML

- On souhaite créer une page HTML qui permet de:
  - Saisir le numéro de l'auteur dans une zone de texte
  - En cliquant sur un bouton, faire appel à un script JSP qui permet de retourner un fichier XML qui contient tous les livres de cet auteur
  - Afficher ces livres dans un TextArea



- 1- Créer la page AjaxXML.htm qui contient les livres d'un auteur donné.
- 2- Créer la page AjaxXML.htm avec des script ajax qui charge le fichier XML et l'affiche

# XmlLivres.php

---

```
<?
$idauteur=$_GET['idauteur'];
$conn=mysql_connect("localhost","root","") or die(mysql_error());
mysql_select_db("db_aj",$conn)or die(mysql_error());
$req="select * from livre where idAuteur=$idauteur";
$rs=mysql_query($req)or die(mysql_error());
header('Content-Type:text/xml');
echo('<?xml version="1.0" encoding="iso-8859-1"?>');
?>
<auteur idAut="<? echo $idauteur ?>">
<? while ($liv=mysql_fetch_assoc($rs)){?>
  <livre
  id="<? echo($liv['id'])?>"
  titre="<? echo($liv['titre'])?>"/>
<? }?>
</auteur>
```

# AjaxXML.htm

```
<html>
<body>
  Id Aut:<input type="text" name="idauteur" id="idauteur" />
  <input type="button" name="ok" value="charger" onclick="go()" />
  <br />
  <textarea name="livres" id="livres" cols="80" rows="5"></textarea>
</body>
<head>
<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("Le navigateur ne supporte pas les objets XMLHttpRequest...");
    xhr = false;
  }
  return xhr
}
```

# AjaxXML.htm (suite)

```
function go(){
var xhr=getXhr();
// On définit ce qu'on va faire quand on aura la réponse
xhr.onreadystatechange=function(){
// si on a tout reçu et que le serveur est ok
if((xhr.readyState==4)&&(xhr.status==200)){
lesLivres=xhr.responseXML.getElementsByTagName("livre");
var txtLivres="";
for(i=0;i<lesLivres.length;i++){
txtLivres+=lesLivres[i].attributes[1].value+"\n";
}
document.getElementById("livres").value=txtLivres;
}
}
// Ici on va voir comment faire une requête post
idauteur=document.getElementById("idauteur").value;
xhr.open("GET","livres.php?idauteur="+idauteur,true);
xhr.send(null);
}
</script>
</head>
</html>
```

# Projet XML et ses applications

---

- On souhaite créer une application web en utilisant la technologie XML et Ajax qui permet de gérer les emplois de temps d'un établissement scolaire.
  - L'établissement est constitué des étudiants.
  - Chaque étudiant appartient à une classe.
  - Chaque classe appartient à une filière
  - Les professeurs assurent des cours des modules à des classes dans des salles à un jour donné, pendant une durée définie par une heure de début et une heure de fin de la séance

# Shémas relationnel

- Le schémas relationnel proposé pour ce projet est le suivant :

## Etudiants

Champ	Type
<u>NUM_INSCRIPTION</u>	varchar(15)
NOM_ET	varchar(25)
PRENOM_ET	varchar(25)
ADRESSE	varchar(70)

## Filières

Champ	Type
<u>ID_FILIERE</u>	int(11)
NOM_FILIERE	varchar(15)
DESCRIPTION	text

## salles

Champ	Type
<u>ID_SALLE</u>	int(11)
NOM_SALLE	varchar(25)
DESCRIPTION	text

## classes

Champ	Type
<u>ID_CLASSE</u>	int(11)
ID_FILIERE	int(11)
NIVEAU	int(11)

## modules

Champ	Type
<u>ID_MODULE</u>	varchar(15)
NOM_MODULE	varchar(25)
DESCRIPTION	text

## professeurs

Champ	Type
<u>ID_PROF</u>	int(11)
NOM_PROF	varchar(25)
TEL	varchar(12)

## cours

Champ	Type
<u>ID</u>	int(11)
ID_CLASSE	int(11)
ID_PROF	int(11)
ID_SALLE	int(11)
ID_MODULE	int(11)
JOUR	varchar(12)
HEURE_DEBUT	time
HEURE_FIN	time

# Travail à faire

---

- 1- Créer et remplir la base de données par des données réelles correspondantes à votre emploi de temps
- 2- Créer un schéma XML qui permet de décrire un emploi de temps sous la forme suivante :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<emploi classe="5SRI">
  <seance jour="lundi" debut="08:30" fin="10:00" prof="A"
    module="M1" salle="lab4"/>
  <seance jour="lundi" debut="10:15" fin="11:45" prof="B"
    module="M2" salle="londres"/>
  .....
  .....
</emploi>
```

# Travail à faire

---

- 3- Créer un script PHP qui permet de générer dynamiquement le fichier XML présentant les données de l'emploi du temps d'une classe donnée.
- 4- Créer un script PHP, utilisant Ajax qui permet de :
  - Sélectionner une classe dans une liste déroulante
  - Après sélection de la classe, afficher l'emploi du temps de cette classe.
- 5- Créer une feuille de style XSL qui permet de transformer le fichier XML en une page WML qui permet de présenter l'emploi du temps d'une classe donnée. Chaque jour de l'emploi sera affiché dans une carte wml.
- 6- Créer une feuille de style XSL qui permet de générer un graphique SVG qui présente pour chaque professeur le nombre d'heures qu'il enseigne.

# Travail à faire

---

- 7- Créer un schéma XML qui permet de décrire la structure du fichier XML qui contient la liste des étudiants d'une classe. La forme de ce fichier est la suivante :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<classe filiere="SRI" niveau="5" >
<etudiants>
  <etudiant numInscription="E200" nom="X" prenom="Y"/>
  .....
  .....
</etudiants>
<modules>
  <module idModule="E200" nomModule="java"/>
  .....
  .....
</modules>
</classe>
```

# Travail à faire

---

- 8- Créer une page PHP qui permet de générer dynamiquement le fichier XML précédent sachant la valeur de ID\_CLASSE
- 9- Créer une feuille de style XSL qui permet de transformer une le fichier XML en un tableau HTML qui présente les étudiants et les modules de la classe.
- 10- Créer une page HTML qui permet de :
  - Sélectionner une classe dans une liste
  - Afficher dans la même page les étudiants et les modules de cette classe.
- 11- Créer une feuille de style XSL qui permet de transformer le document XML précédent en un document SMIL qui permet de faire une présentation multimédia des étudiants d'une de cette classe.
- 12- Créer page en utilisant Ajax, XML et PHP pour la saisie d'un nouveau emploi du temps.
- 13- Proposer une page principale dans laquelle, vous regroupez toutes les fonctionnalités que vous avez développé dans ce projet.
- 14-Préparer un rapport de projet
- 15-Envoyer le projet avant le 15 janvier 2008 à l'adresse [estem@youssfi.net](mailto:estem@youssfi.net)

# Frameworks Ajax

---

- Pour éviter à un développeur Ajax d'écrire beaucoup de code java script, de nombreux frameworks javascript intégrant Ajax ont été développés. Exemple :
  - Spry de Adobe
  - Rico
  - Prototype
  - Google
  - Yahoo
  - ....
- Dans cette partie, nous allons nous intéresser à Spry.

# Framework Spry

---

- Le cadre applicatif Spry 1.4 pour Ajax est une bibliothèque JavaScript qui permet aux concepteurs Web de créer des pages Web offrant une expérience enrichie aux visiteurs de leurs sites.
- Avec Spry, vous pouvez utiliser du codeHTMLet CSS, ainsi qu'une quantité minime de JavaScript, afin
  - d'incorporer des données XML dans vos documents HTML,
  - de créer des widgets tels que des accordéons et des barres de menus, ou encore ajouter différents effets à divers éléments de page.
- Le cadre applicatif Spry est conçu de telle sorte que le code soit simple et facile à utiliser pour toute personne possédant une connaissance de base du langage HTML, de CSS et de JavaScript.

# Composants de Spry

---

- Le cadre applicatif Spry 1.4 comprend trois composants principaux qui permettent de créer des pages dynamiques :
  - Les widgets,
  - les ensembles de données XML
  - et les effets.
- Les widgets sont des éléments de page, comme des accordéons et des panneaux à onglets, qui accroissent l'attrait et l'interactivité d'une page.
- Les ensembles de données XML permettent d'afficher sur la page Web des données provenant d'une source de données XML,
- Enfin, les effets Spry, tels que Fondu ou Ecraser, ajoutent du mouvement à la page, de manière à améliorer l'expérience de l'utilisateur.
- vous pouvez afficher des données XML à l'intérieur d'un widget et lui ajouter des effets afin de créer des pages plus riches que ce que permet le code HTML statique.

# Intégrer Spry à votre projet

---

## ○ Préparation des fichiers

- Téléchargez et liez les fichiers appropriés.
- Téléchargez le fichier ZIP de Spry sur le site Adobe® Labs. et décompressez-le sur votre disque dur.
- Pour intégrer la totalité du framework, copiez le dossier Spry à l'intérieur du dossier de votre projet

# Utilisation des widgets Spry

---

- Un *widget Spry* est un élément de page qui combine du code HTML, CSS et JavaScript pour permettre l'interaction de l'utilisateur.
- Le widget Spry se compose des éléments suivants :
  - **Structure du widget** Bloc de code HTML qui définit la composition structurelle du widget.
  - **Comportement du widget** Du code JavaScript qui détermine comment le widget répond aux événements provoqués par l'utilisateur.
  - **Style du widget** Du code CSS qui définit l'apparence du widget.

# Utilisation des widgets Spry

---

- Les widgets intégrés dans le framework Spry 1.4 sont :
  - **Accordéon**
  - **Panneau réductible**
  - **Panneaux à onglet**
  - **Barre de menus**
  - **Champ de texte de validation**
  - **Zone de texte de validation**
  - **Validation de la sélection**
  - **Validation de case à cocher**