Asynchronous **J**avaScript **A**nd **X**ML
AJAX

C. Petitpierre

2.3.2009

Des exemples et le code de l'appel AJAX complet sont disponibles en http://lti.epfl.ch/AJAX

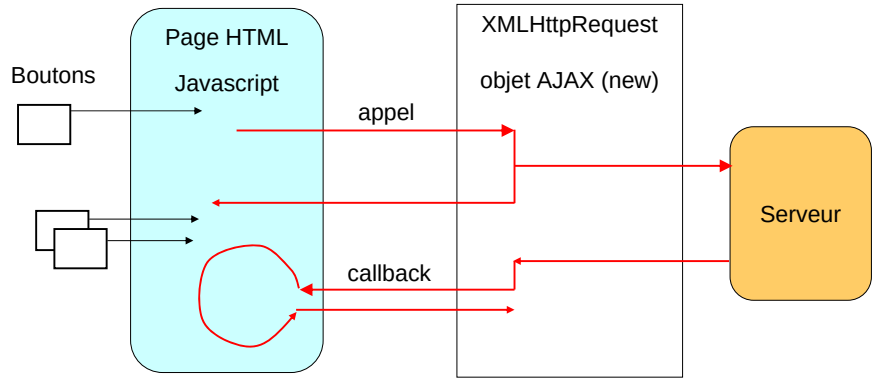
Faculté I&C, Claude Petitpierre, André Maurer

AJAX

- AJAX est un type de programmation rendu populaire en 2005 par Google.
- AJAX n'est pas un nouveau langage de programmation, mais une nouvelle façon d'utiliser les standards existants.
- AJAX est basé sur JavaScript et HTTP.
- LemanOS est construit sur AJAX, avec quelques couches standards supplémentaires

Faculté I&C, Claude Petitpierre, André Maurer

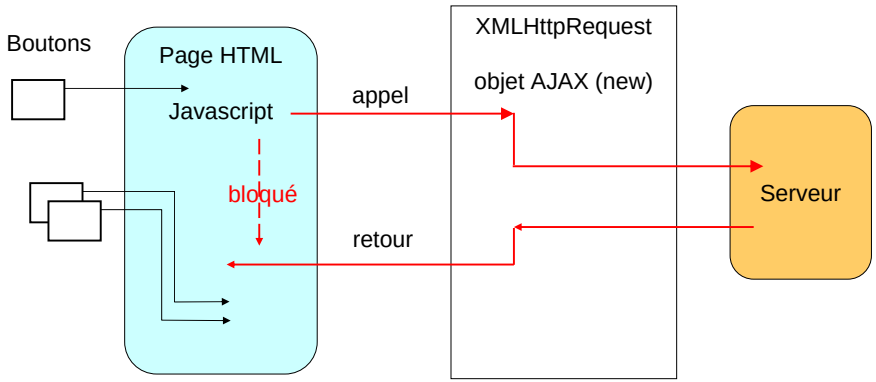
Appel AJAX aynchrone



Le callback fonctionne exactement comme le timeout.
Il appelle une fonction désignée par l'appelant.

Faculté I&C, Claude Petitpierre, André Maurer

Appel AJAX synchrone



L'exécution de l'envoi fonctionne exactement comme un appel de méthode simple.

Côté serveur

Commande HTTP

http://epfl.ch/course/view.php?id=376

- adressant des fichiers HTML
- des cgi, des servlets, des asp (Microsoft) ou …

L'appelant ne peut pas savoir par l'analyse des messages si ce sont des fichiers qui sont retournés ou des programmes qui répondent

Côté navigateur

- On peut traiter la réponse en l'interprétant comme une structure XML:

```
http_request = new XMLHttpRequest()
   . . .
xml = http_request.responseXML
xml.getElementsByTagName("aaaa")[0]
                        .firstChild.nodeValue
```

ou comme …

Côté navigateur

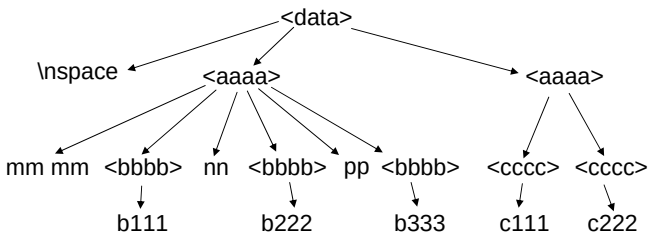
- ou comme une structure Text:

```
http_request = new XMLHttpRequest()
   . . .
x = http_request.responseText
document.getElementById("Display")
                        .innerHTML = x
```



```
<?xml version='1.0' encoding='UTF-8'?>
<data>
  <aaaa>
    mm mm
    <bbbb>b111
    </bbbb>
    nn
    <bbbb>b222
    </bbbb>
    pp
    <bbbb>b333
    </bbbb>
  </aaaa>
  <aaaa>
    <cccc>c111
    </cccc>
    <cccc>c222
      . . .
```

Arbre XML



Lecture d'un arbre XML

```
<?xml version='1.0' encoding='UTF-8'?>
<data>
   <aaaa>
      mmmm
       <bbbb>b111
      </bbbb>
      <bbbb>b222
      </bbbb>
      <bbbb>b333
      </bbbb>
   </aaaa>
   <aaaa>
      <cccc>c111
      </cccc>
      <cccc>c222
         . . .
```

```
xml.getElementsByTagName("aaaa")[0]
                .firstChild.nodeValue: mmmm
xml.getElementsByTagName("bbbb")[1]
                .lastChild.nodeValue: b222
xml.getElementsByTagName("aaaa")[0]
                .getElementsByTagName("bbbb")[2]
                .firstChild.nodeValue: b333
```

http://lti.epfl.ch/AJAX

Affichage de l'arbre XML

```
function print(indent, t) {
   if (t.nodeValue != undefined) {        // soit affiche un nœud
      txt.push(indent + t.nodeValue)
   } else {                               // soit parcourt les enfants
      txt.push(indent +"\240--\240"+ "&lt;" + t.tagName +"&gt;")
      for (var j=0; j<t.childNodes.length; j++) {
         print(indent+"\240\240\240\240", t.childNodes[j])
} } }
```

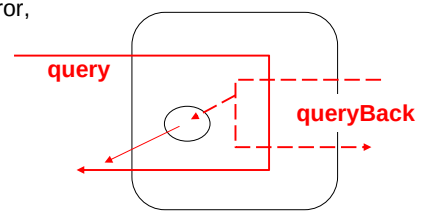

Asynchronous call

```
function makeRequest(type, alertFunction) {
      //  . . .
      http_request.onreadystatechange = function() {
       alertFunction(http_request)  // calls the user-defined function
   }
   http_request.open('GET', URL, true)
   http_request.send(null)
}

// user-defined function
var alertContents = function (http_local_request) {
        document.getElementById("Display").innerHTML
                        = http_local_request.responseText
}

makeRequest("text.html", alertContents)
// On return, calls alertContent() to display returned value
```

asynchronous

Appels synchrone - asynchrone

```
   // . . .
   DataBaseService.query(database.readyQuery,{
      callback: database.queryBack,
      errorHandler:database.handleDBError,
      async: false
   })
   if (database.error != null) {
      throw database.error
   }
   return database.data
}
/**
 * Call back function, just stores the result for the synchronous function
 */
database.data = null
database.queryBack = function (data) {
   database.data = data
}
```



Création d'un arbre XML

Ecrire un arbre est beaucoup plus facile que le lire, il suffit de concaténer les éléments de texte:

```
txt = [ ]
txt.push("<?xml version='1.0' encoding='UTF-8'?>")
txt.push("<data>")
txt.push("<aaaa>" + uneValeur)
    . . .
File.write("toto.xml", txt.join("\n"))
```