

Université
Nice
Sophia Antipolis

Programmation Web - Client riche

Gaëtan Rey
Gaetan.Rey@unice.fr
DUT Informatique – 2016

CC BY NC ND

Université
Nice
Sophia Antipolis

Objectifs

- ▶ Introduction / Rappel sur le web
- ▶ Navigateurs et DOM
- ▶ JavaScript
- ▶ Requêtes asynchrones et formats d'échanges de données
- ▶ Exemples de bibliothèques et autres langages

Janvier 2016 Gaëtan Rey – Université Nice Sophia Antipolis 2

Université
Nice
Sophia Antipolis

REQUÊTES ASYNCHRONES ET FORMATS D'ÉCHANGE DE DONNÉES

Janvier 2016 Gaëtan Rey – Université Nice Sophia Antipolis 67

Université
Nice
Sophia Antipolis

Asynchronous JavaScript and XML

- ▶ AJAX n'est en réalité qu'une compilation de différentes technologies pour construire des applications Web et des sites web dynamiques interactifs
 - ▶ JavaScript, CSS, JSON, XML, le DOM et le XMLHttpRequest
- ▶ Ajax vise à :
 - ▶ diminuer les temps de latence,
 - ▶ éviter le rechargement de la page,
 - ▶ apporter de nouvelles fonctionnalités et
 - ▶ augmenter la réactivité de l'application Web.
- ▶ Les applications Web qui possèdent ces qualités sont appelées Rich Internet Applications

[\[Wikipédia\]](#)

Janvier 2016 Gaëtan Rey – Université Nice Sophia Antipolis 68

Université
Nice
Sophia Antipolis

Modèle classique et modèle AJAX

Modèle Classique

Janvier 2015 Gaëtan Rey – Université Nice Sophia Antipolis 69

Université
Nice
Sophia Antipolis

XMLHttpRequest (1)

- ▶ Initialement un composant ActiveX créé en 1998 par Microsoft pour leur application web Outlook Web Access
- ▶ Ajouté à la norme ECMAScript relative au langage JavaScript et mis en œuvre sur la plupart des navigateurs du marché entre 2002 et 2005
- ▶ Permet d'effectuer des requêtes asynchrones sur un serveur
 - ▶ Attention, il faut un serveur http, sinon ça ne fonction pas !!

[\[Wikipédia\]](#)

Janvier 2016 Gaëtan Rey – Université Nice Sophia Antipolis 70

XMLHttpRequest (2)

- Quelques propriétés
 - onreadystatechange** : Spécifie la fonction à appeler lorsque la propriété `readyState` varie
 - readyState** : L'état d'avancement de la requête
 - 0 (uninitialized) -> non initialisé
 - 1 (loading) -> début du transfert des données
 - 2 (loaded) -> données transférées
 - 3 (interactive) -> les données reçues sont partiellement accessibles
 - 4 (complete) -> les données sont complètement accessibles
 - responseText** : La réponse à la requête (lecture seule) sous la forme d'une chaîne de caractères
 - responseXML** : la réponse à la requête (lecture seule) sous la forme d'un objet XML
 - status** : le code HTTP retourné par la requête

XMLHttpRequest (3)

- Quelques méthodes
 - abort()** : Annule la requête courante
 - getAllResponseHeaders()** : Retourne les noms et les valeurs de tous les en-têtes HTTP sous forme d'une chaîne de caractères
 - getResponseHeader(header)** : Retourne la valeur d'un de l'en-tête HTTP *header* sous forme d'une chaîne de caractères
 - open(method, url[, asynchrone[, user[, password]]])** : Initialise une requête en spécifiant la méthode de connexion, l'URL, si le mode est asynchrone et en indiquant d'éventuelles informations d'identification

XMLHttpRequest (4)

- Quelques méthodes
 - send(data)** : Envoie la requête HTTP au serveur en transmettant éventuellement des données
 - setTimeouts(timeout)** : Définit la durée *timeout* maximale accordée à une requête pour quelle s'effectue complètement
 - setRequestHeader(nom, valeur)** : Définit un en-tête HTTP (couple nom=valeur) à envoyer avec la requête

XMLHttpRequest Un premier exemple

```
function initXhr(){
  if (window.XMLHttpRequest) // tous les navigateurs modernes
    xhr = new XMLHttpRequest();
  else if (window.ActiveXObject) // ancienne version d'Internet Explorer
    xhr = new ActiveXObject("Microsoft.XMLHTTP");
  else // XMLHttpRequest non supporté par le navigateur
    console.log("Votre navigateur ne supporte pas XMLHttpRequest...");
}

xhr.onreadystatechange = function() {
  if (xhr.readyState == 4 && xhr.status == 200) // xhr.status == 0 si ftp ou file
    maCallback(xhr.responseText); // ou maCallback(xhr.responseXML);
};

xhr.open(protocol, url, asynch);
xhr.send(data);
```

Envoyer des données

- Dans le cas de GET


```
var val1 = encodeURIComponent("ma valeur");
xhr.open("GET", "fichier.php?var1="+val1+"&var2=val2&var3=val3", true);
xhr.send(null);
```
- Dans le cas de POST


```
xhr.open("POST", "fichier.php", true);
xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
xhr.send("var1=val1&var2=val2&var3=val3");
```

Requêtes multiples

- Attention,
 - à ne pas modifier les paramètres durant l'exécution d'une requête
 - à ne pas envoyer une requête plusieurs fois
- Solution
 - N'utiliser qu'un seul objet XMLHttpRequest
 - Ne pas envoyer de requête si `readyState != 0`

Les données reçues

3 grands formats de données :

- Texte
 - xhr.responseText
- XML
 - xhr.responseXML
- JSON
 - xhr.responseJSON

A gérer comme des objets JavaScript, via la fonction eval()

JSON : La syntaxe

Les éléments de JSON sont :

- objet** :
 - Contient une liste de membre (objet ou variable)
 - { membre, membre, ... }
 - Un membre
 - "nom" : "valeur"
- variable scalaire**
 - Number, String, Boolean
- tableau**
 - [valeur, valeur,]
- valeur littérale**
 - null, true, false, chaîne de caractères, et les valeurs numériques

Traitement des données

```

xhr.onreadystatechange = function() {
  if (xhr.readyState == 4 && xhr.status == 200)
    maCallback(xhr.responseText); // on récupère du texte
};

function maCallback(data) {
  var doc = eval('(' + data + ')'); // transforme le texte en objet JavaScript

  var nomMenu = doc.menu.value; // assigner une valeur au champ

  doc.commandes[0].titre; // lire la valeur de "titre" dans le tableau
  doc.commandes[0].action; // lire la valeur de "action" dans le tableau
}
  
```

JSON vs. XML

Les avantages de JSON :

- La vitesse de traitement (code compact)
- La simplicité de mise en œuvre
 - Idéal pour la sérialisation d'objet
- Les contenus binaires peuvent être intégrés et échangés via une représentation textuelle spéciale
 - new Buffer(file).toString('base64');

Les avantages de XML :

- XML est extensible
- Proche des formats comme RSS, SVG
- Largement utilisé et reconnu par tous les langages de programmation
- Plus facile à lire et convient mieux pour les fichiers destinés aux non développeurs

XML et JSON ne conviennent pas pour stocker directement des données binaires de taille importante

XMLHttpRequest Cross-Domain

La notion de cross-domain consiste à faire une requête sur un autre domaine que le domaine d'origine de la page

Les navigateurs supportent tous XMLHttpRequest dans sa *spécification Level 1* qui ne permet pas de faire du cross-domain

Le W3C travaille sur la *spécification Level 2* qui autorise le cross-domain

XMLHttpRequest Cross-Domain

2 solutions

```

function initXdr() {
  if (window.XDomainRequest)
    xdr = new XDomainRequest();
  else if (window.XMLHttpRequest)
    xdr = new XMLHttpRequest();
  else
    console.log("Votre navigateur ne supporte pas le cross domaine...");
}

xdr.onload = function() { // gestion simplifiée, plus besoin de tester status, ...
  maCallback(xdr.responseText);
}

xdr.open("GET", "http://autredomaine");
xdr.send();
  
```

Université Nice Sophia Antipolis

Le minimum à connaître

- ▶ La notion d'appels asynchrones
- ▶ L'objet XMLHttpRequest
- ▶ La description des données en XML et JSON
- ▶ La notion de domaine et d'appels cross-domain

Janvier 2016 Gaëtan Rey - Université Nice Sophia Antipolis 83

Université Nice Sophia Antipolis

Quelques liens utiles

- ▶ Google Hosted Libraries
 - ▶ <https://developers.google.com/speed/libraries/>
- ▶ Microsoft Ajax Content Delivery Network
 - ▶ http://www.asp.net/ajax/cdn#jQuery_Releases_on_the_CDN_0
- ▶ Mozilla Developer Network
 - ▶ <https://developer.mozilla.org/fr/>
- ▶ Developpez.com
 - ▶ <http://web.developpez.com>

Janvier 2016 Gaëtan Rey - Université Nice Sophia Antipolis 120

Université Nice Sophia Antipolis

Bibliographie / Remerciement

- ▶ Merci
 - ▶ A la communauté Developpez.com
 - ▶ A la communauté Wikipédia
- ▶ Cours sur JavaScript utilisés pour monter ce module
 - ▶ [Introduction au JavaScript](#) de Serge P.
 - ▶ [Développement Web : « Zone Grand Débutant »](#) de Guillaume Rossolini
 - ▶ [Cours de JavaScript](#) de Jacques Guizol
 - ▶ [Apprendre le JavaScript](#) de Van Lancker Luc
 - ▶ [Syntaxe JavaScript](#) de [Wikipédia]
 - ▶ [JavaScript](#) sur le Mozilla Developer Network
- ▶ Autres supports utilisés
 - ▶ [Comment fonctionnent les navigateurs](#)

Janvier 2016 Gaëtan Rey - Université Nice Sophia Antipolis 121