IUT- Département d'Informatique
G. Rey
2016

Programmation Web – client riche
M4103 - TD n° 4 (séance 5 et 6)

CC BY NC ND

1 Objectifs

Ce TD illustre la partie du cours sur la programmation web et en particulier les concepts liés à la conception d'application riches coté client, c'est-à-dire dans le navigateur. Les concepts principaux abordés seront :

- Les principes avancés du langage JavaScript
- L'exploitation du DOM
- AJAX et XML
- Mise en œuvre d'une application en JavaScript

Un compte rendu donnant la réponse à chacune des questions posées devra être rendu à votre responsable de TD par email à la fin de chacune des séances. L'objet (sujet) de l'email devra être le suivant :

[S4T][JS][Gx][TDx]Nom-Prénom / Nom-Prénom

Avec Gx le numéro de votre groupe (exemple G1 pour le groupe 1) et TDx le numéro du TD inscrit sur le sujet du TD (exemple TD1 pour le premier TD).

Vous ne pouvez pas faire 2 séances avec le même binôme. De même, vous ne pouvez pas faire plus de la moitié des séances seul sauf si votre groupe de TD est assez petit pour que chaque étudiant soit sur une machine de l'IUT.

Veuillez, sauf indication contraire de votre responsable de TD, faire l'ensemble des exercices marqués « Obligatoire » dans l'ordre de la feuille de TD. Les exercices « Conseillés » sont à faire si vous avez terminé les précédents ou lors de vos révisions. Les exercices « optionnels » sont là pour les plus passionnés d'entre vous.

Attention, il faut lire l'exercice en entier avant de vous lancer dans sa réalisation.

2 Documentation (obligatoire)

Pour ce TD mais également pour les TD suivant, sauf mention contraire dans l'énoncé ou si un lien vous est fourni, vous ne devez pas aller chercher des réponses sur internet !! Les seuls sites suivants sont autorisés :

- Le site contenant les supports de cours et de TD.
- Le site de Mozilla : Mozilla Developer Network.
- Les sites du W3C : pour la validation HTML5 ou du CSS. Mais aussi pour avoir la documentation sur le DOM.

3 Prise en main de l'environnement (obligatoire)

Pour réaliser ce TD, vous aurez également besoin d'un éditeur de texte pour écrire le code de vos pages. Vous pouvez par exemple utiliser un des logiciels Notepad++, ConText, Quanta+, WebExpert,... Microsoft vous propose de télécharger gratuitement Expression Web.

Pour ce TD, nous aurons besoin de faire fonctionner nos pages sur un véritable serveur web, capable de répondre à nos requêtes http et d'interpréter des scripts PHP (coté serveur). Pour cela nous utiliserons le serveur mis à votre disposition par le département informatique de l'IUT. Les informations nécessaires sont disponibles sur cette page.

Pour déposer vos fichiers sur l'espace web, vous pouvez utiliser soit votre partage de fichiers (p:\web\) depuis les machines du département, soit une connexion ssh, scp ou sftp via différents outils (WinSCP, FileZilla, Cyberduck, Core FTP LE, ...) vers le server linserv1 (depuis le réseau interne) soit lindmz.unice.fr depuis l'extérieur.

Programmation Web – client riche
M4103 - TD n° 4 (séance 5 et 6)

cc BY NC ND

Dans votre compte rendu, vous devrez donc d'une part rendre l'ensemble des fichiers (html, css et js) que vous avez écrit. Mais également le lien sur votre travail. Le lien aura la forme suivante : http://linserv1/users/etudiant/x/xy123456/web/. De fait, vous ne devez pas utiliser un autre serveur web que celui qu'on vous fournit.

Pensez à sauvegarder régulièrement votre travail et à commenter votre code. N'hésitez pas à sauvegarder les différentes versions, évolutions d'un même exercice de manière à pouvoir facilement réviser ou le réutiliser par la suite.

4 Exercice 1 : 2048 (obligatoire)

Si vous n'avez pas fini l'exercice 2 du TD précédent, vous avez **1h30 maximum** pour le terminer. Ensuite vous devrez passer à l'exercice suivant de ce TD et vous devrez finir les exercices obligatoires de TD précédent chez vous d'ici la prochaine séance.

Attention, la qualité du code produit pour l'exercice sur 2048 sera évaluée. Pensez à nommer vos variables convenablement et commenter et indenter votre code. L'ensemble des identifiants des constantes sera regroupé en début de fichier pour facilement être modifier ces dernières en cas de besoin. Et bien entendu, on ne retrouvera aucune valeur de constante dans le code.

5 Exercice 2 : RSS Reader (obligatoire)

RSS (sigle venant de l'anglais « Really Simple Syndication ») est une famille de formats de données utilisés pour la syndication de contenu Web.

Un produit RSS est une ressource du World Wide Web dont le contenu est produit automatiquement (sauf cas exceptionnels) en fonction des mises à jour d'un site Web. Les flux RSS sont des fichiers XML qui sont souvent utilisés par les sites d'actualité et les blogs pour présenter les titres des dernières informations consultables. [Wikipédia]

5.1 Mise en place basique du protocole de communication asynchrone

- Commencez par écrire une page web (HTML5 et CSS) qui proposera de choisir un flux dans une liste. Pour le moment la liste sera composée des trois noms. Chaque nom sera associé à une des URLs suivantes : La SIF, l'université Nice Sophia Antipolis et les actualités pro de Clubic. Cette liste pourra prendre différentes formes (select, ul/li, …), à vous d'être créatif. Une zone de type DIV pourra être ajoutée pour faciliter les insertions à venir.
- Ajoutez à votre fichier de script les fonctions suivantes :
  -- init() : on chargement de la page elle crée un objet de type XMLHttpRequest accessible à toutes les autres fonctions
  -- selectionRSS(valeur) : quand l'utilisateur sélectionne un flux RSS dans la liste, on s'abonne au changement d'état de l'objet XMLHttpRequest. Et quand on a bien reçu le document, on appelle la fonction de traitement (maCallback()). Il faut aussi préparer la requête http asynchrone de type GET et l'envoyer.
  -- maCallback(data) : insert simplement les données de type texte dans la page html.

- Quelle erreur obtenez-vous ?
- Pourquoi la solution ci-dessus de fonctionne pas ?

5.2 Les fichiers locaux

Pour résoudre ce problème nous allons commencer par mettre en place une première solution simple.

- Enregistrez une copie locale des trois fichiers XML précédent.
- Modifiez votre code pour lire les fichiers locaux. Les fonctions modifiées seront suffixées par un 2 (exemple init2()), de manière à conserver les fonctions de chaque version.
- Vérifiez que cette fois tout fonctionne correctement.

Programmation Web – client riche
M4103 - TD n° 4 (séance 5 et 6)

BY NC ND

5.3 Proxy php

La solution précédente nous a permis de valider le fonctionnement de notre code, mais celle-ci n'est pas satisfaisante. Une solution, sans faire de cross-domaine et d'utiliser un petit proxy en PHP. Voici le code du proxy que nous utiliserons pour lire les fichiers RSS et les transmettre à notre client web.

```
<?php
	$url = $_POST["rssURL"];
	if ($url != ""){
		$monRSSDoc = new DOMDocument();
		if ($monRSSDoc->load($url)){
			header('Content-Type: text/xml');
			echo $monRSSDoc->saveXML();
		}else{
			echo "erreur de lecture";
		}
	}else{
		echo "URL vide";
	}
?>
```

Vous n'avez pas le droit de modifiez ce fichier proxy.php.

- Enregistrez le code ci-dessus dans un fichier proxy.php
- Modifiez votre code pour qu'il appelle le proxy en lui passant en paramètre l'url du flux recherché. Les fonctions modifiées seront suffixées par un 3 (exemple init3()), de manière à conserver les fonctions de chaque version.
- Vérifiez que tout fonctionne toujours correctement.

5.4 Analyse du contenu et mise en page

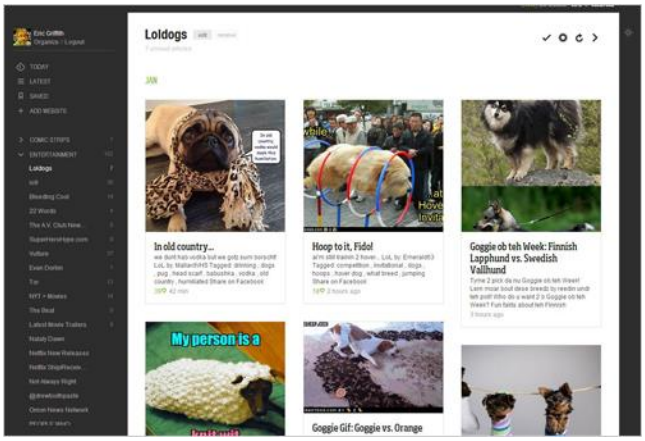
Puisque nous recevons un fichier RSS, il y a probablement mieux à faire que d'afficher bêtement du texte. Déjà, nous savons qu'il s'agit d'un fichier XML. Mais en plus de cela nous savons qu'il respecte la DTD décrivant les flux RSS. Nous allons donc mettre en place script capable d'analyser ces documents et de nous les présenter de manière plus agréable dans notre page. Pour cela nous nous reposerons sur la description des balises présente sur cette page Wikipédia.

- Ajoutez à votre page une présentation des métadonnées suivantes (si celles-ci sont disponibles) : titre, description, lien et date de publication. A loisirs vous pouvez également ajouter les 4 autres métadonnées.
- Ajoutez à votre page un affichage dédié pour chaque article. On devra avoir un affichage simple mais compact présentent qu'une partie des informations. Et un affichage détaillé permettant d'avoir accès à l'intégralité des données contenus dans le flux rss.
- Augmentez la liste de liens rss disponible avec 2 ou 3 autres liens de votre choix.
- Ajoutez un champ texte permetant à l'utilisateur de saisir l'url du flux rss si celui-ci n'est pas disponible dans votre liste.

Programmation Web – client riche
M4103 - TD n° 4 (séance 5 et 6)

CC BY NC ND

Voici quelques illustrations pour vous donner des idées de ce que vous pouvez faire :



5.5 Et le Json ?

Pour tester l'exploitation des données en json, nous allons simplement demander à notre proxy de convertir les données dans ce format avant de nous les transmettre.

- Modifiez votre code pour qu'il utilise proxyJson.php.
- Modifiez le traitement des données qui sont maintenant en json (texte) et non plus en XML.

```
<?php
	$url = $_POST["rssURL"];
	if ($url != ""){
		$monRSSDoc = new DOMDocument();
		if ($monRSSDoc->load($url)){
			$xml = simplexml_load_string($monRSSDoc->saveXML());
			echo json_encode($xml);
		}else{
			echo "erreur de lecture";
		}
	}else{
		echo "URL vide";
	}
?>
```

5.6 Le concurrent : Atom (conseillé)

Un document au format Atom est appelé un « fil de syndication Atom » ou fil Web. Ces fils peuvent être affichés aussi bien sur un site Web que directement dans un agrégateur, qui est un logiciel prévu à cet effet. Cela permet de suivre, ou «

Programmation Web – client riche
M4103 - TD n° 4 (séance 5 et 6)

BY NC ND

s'abonner », à un fil. Le propriétaire d'un site web peut quant à lui utiliser un logiciel spécialisé, tel qu'un système de gestion de contenu, pour publier une liste de ressources, dans un format standardisé et lisible par une machine, et dont il souhaite notifier des mises à jour.

Le développement d'Atom a été justifié par le manque de flexibilité commun aux nombreuses variantes de RSS. [Wikipédia]

- Modifiez votre code pourqu'il supporte en même temps les documents de type Atom 1.0 et RSS 2.0. Les correspondances entre les balises sont disponibles sur cette page.

6 Conclusion

J'espère que ce quatrième TD vous a permis de découvrir ou redécouvrir la partie liée au requêtes asynchrone en JavaScript ainsi que le lien avec le traitement de fichier XML et vous a donné une idée des possibilités qu'offre ce langage.