Bénet, Lang, Guyonnet, Vargas, Abbasi

Projet Tuteuré 2015

Résumé Intelligent des Nouvelles du Jour

Projet Tuteuré 2015

N° du sujet : 223

Titre du sujet : Résumé 'intelligent' des nouvelles du jour

Nom des membres du groupe : ABBASI Antoine, BENET Baptiste, GUYONNET Bastien, LANG David, VARGAS Clément

* Résumé du Sujet

Ce projet consiste à développer une application en ligne permettant de récupérer les actualités et de les transmettre à un utilisateur, le tout en gérant les redondances des informations.

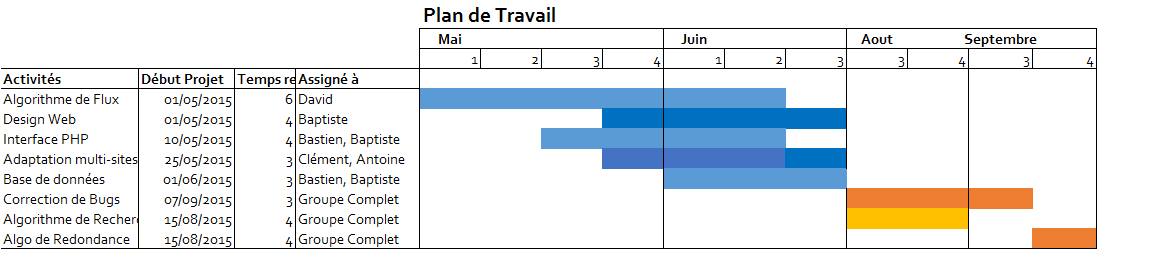
Nous avons choisi de rajouter une option de connexion pour les utilisateurs, qui pourront enregistrer leurs catégories favorites.

Une autre question est aussi celle de la pertinence des articles, les critères que nous avons sélectionnés sont arbitraires et aussi choisis en fonction de leur popularité.

# Répartition des Rôles au sein du groupe

Nous nous sommes répartis les rôles par préférence.

Baptiste et Bastien se sont occupés de la structure du site, de sa mise en page et de la gestion des données sur le site.

Tandis que David et Clément se sont occupés de la récupération des données.

Nous avons tenté de répartir les tâches de manière à ne pas ralentir certaines tâches qui nécessitaient la fin d’une autre. C’est pourquoi nous avons choisi de diviser en deux le travail, d’un coté le coté « design » réalisé par Bastien et Baptiste, et de l’autre la partie plus « technique » nécessitant plus de réflexion dans sa mise en place.

Notre travail en Javascript

Au cours du semestre 4, nous avons appris à utiliser le Javascript , avec lequel nous avons appris à récupérer des contenus RSS.

- Dans le cadre de note projet nous utilisions auparavant le langage PHP, notamment CURL (expliqué dans notre premier rapport traitant du contenu PHP). Mais après avoir étudié le Javascript au cours de ce semestre, nous avons choisi d’utiliser ce langage et de modifier entièrement notre projet.

# Pourquoi avons-nous choisi le Javascript ?

Nous nous sommes rendu compte que le PHP n’était pas du tout optimisé pour traiter des flux de données et avons dû passer par CURL, simulant plus ou moins une utilisation par le Javascript.

Nous ne connaissions pas le langage Javascript avant de commencer note dernier semestre, et nous avions déjà appris le php. Cependant, en découvrant le Javascript, nous nous sommes rendu compte qu’il était bien plus adapté à nos besoins, et permettait notamment la récupération et le traitement de flux RSS. L’accès aux flux RSS nous permet d’automatiser la récupération des données (tandis que nous récupérions les pages en cherchant manuellement en php).

# Utilisation du RSS pour le traitement des données

Après avoir étudié différentes méthodes de traitement de l’information, (que ce soit en PHP ou en Javascript), nous avons choisi d’utiliser les flux RSS. Simplement car c’est un format optimisé pour l’actualisation, ce qui convenait parfaitement à notre sujet.

# Utilisation d’Ajax dans la récupération des données

Les requêtes AJAX nous permettent d’échanger avec les serveurs des journaux dont on souhaite récupérer les flux rss. Les données reçus sont sous la forme XML, que l’on doit ensuite parser. L’utilisation d’un proxy CORS est nécessaire pour autoriser la réponse d’un serveur étranger.

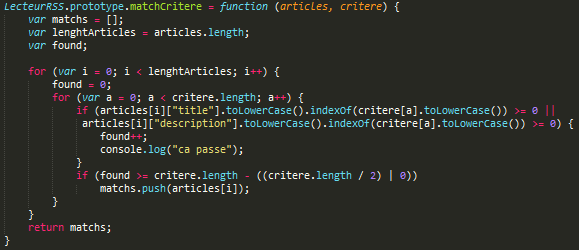


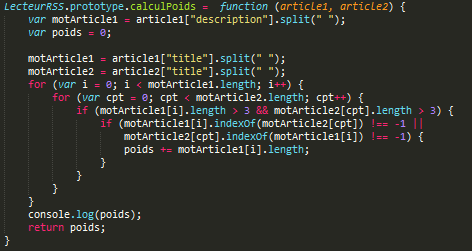
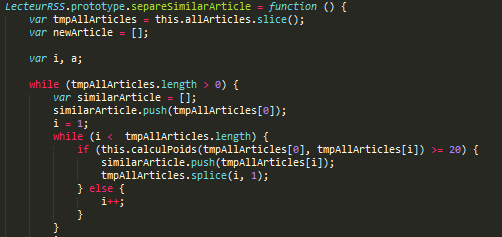
# 

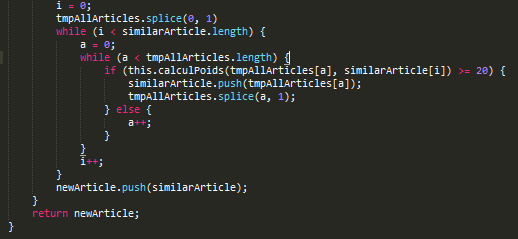
# Traitement des données

Avant d’afficher les données parsées, il faut d’abord vérifier que certains articles ne portent pas sur le même sujet. Pour faire cela, on compare les titres des articles en vérifiant que chaque mot du titre d’un article ne se trouve pas dans un autre titre. Si c’est le cas, on incrémente de la taille du mot une variable de « poids » (seuls les mots de plus de 3 caractères sont pris en compte dans la comparaison, puisqu’une grande majorité des mots de taille inférieure sont souvent simplement des mots de liaison). On remplit un tableau d’articles similaires et on supprime les articles trouvés dans le résultat de la recherche. On répète une deuxième fois avec tous les articles similaires, on remplit le tableau d’articles similaires et on supprime une fois de plus les articles trouvés dans le résultat de la recherche.

Pourquoi répète-t-on 2 fois la procédure ? **Parce qu’un premier article similaire à un second et ce second similaire à un troisième n’implique pas que le premier et le troisième sont semblables.**







# Explication en détail : Ordre d’exécution du script

* **LecteurRSS.main** = initialise l'évènement sur le click du bouton de recherche.
* **LecteurRSS.search** = lance la recherche, appelle la méthode send pour chaque flux RSS.
* **LecteurRRSS.send** = appelle asynchrone qui requête un flux RSS.
* **ReaderRSS.parseRSS** = appelle la méthode parseArticle pour tous les articles trouvés, récupère l'article renvoyé par parseArticle et le met dans une liste, puis renvoie la liste à la fin.
* **ReaderRSS.parseArticle** = lit l'article ("titre", "link", "description", "pubdate" etc.), renvoi un Array (qui est l'article ) contenant toute les information de l'article.
* **LecteurRSS.matchCritere** = supprime les articles qui ne correspondent pas aux critères de recherche
* **LecteurRSS.separeSimilarArticle** = Regroupe les articles similaires ensemble.
* **WriterRSS.writeRSS** = Parcoure la liste de list d'article similaire et appelle pour chacune la méthode writeArticle et envoie l'index de l'article de la liste à écrire, celui préféré par l'utilisateur (Le Monde, 01net etc.), c’est-à-dire qui cherche un article qui convient aux critères de préférence et si aucun ne correspond alors on prend le premier de la liste d'articles similaires.
* **WriterRSS.writeArticle** = créer des nœuds DOM et les remplis avec l’information de l'article, ajoute si disponible les liens de tous les autres articles de la Liste d'article similaires
* **WriterRSS.addEvent** = ajoute la gestion de play/pause pour video/audio en authorisant une seule lecture à la fois si jamais l'article contient un media.

****

Dans cet affichage, nous pouvons voir qu’il existe des articles similaires au premier sélectionné et que l’on a donc correctement réussi à effectuer un tri des retours à afficher pour l’utilisateur afin qu’il n’ai qu’un seul article visible.

# Design et conception du site web

Nous avons opté pour une interface épurée et facile à lire.

* Une barre de recherche
* Une selection de sites web d’où l’on récupère les informations
* Un affichage des flux RSS optimisé pour le confort de lecture.

# C:\Users\Baptiste Bénet\AppData\Local\Microsoft\Windows\INetCache\Content.Word\NewsFeedm1.pngPremière version du site

# Version Plus épurée



# Version quasi finale du site web avec un aspect plus professionnel



Comme on peut le constater, le site dispose d’une interface plus épurée, la recherché se veut simple et rapide d’accès afin de prioriser le retour de l’information.

# Conclusion

Ce projet nous a permis de prendre du recul et de former une petite entreprise dans laquelle nous devions agir en tant que professionnels. Nous avons donc pu avoir un véritable aperçu du travail d’entreprise.  
Chacun de nous a des forces et des faiblesses très différentes. Pour nous l’une des difficultés a été de faire coïncider celles-ci pour que l’on soit tous capables de fournir une charge de travail équivalente. Il nous donc fallut répartir correctement les tâches.

* Nous avons beaucoup appris de ce projet techniquement parlant et avons dû apprendre à faire des recherches par nous même sans l’appui d’un professeur pour nous guider.
* Nous nous sommes rendus compte qu’il est plus facile d’avancer dans son projet si l’on a un planning clairement établi définissant les tâches à effectuer.

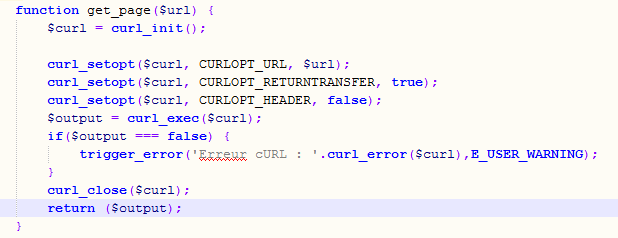
Nous remercions Monsieur Champin qui a accepté de nous assister durant ce semestre et nous espérons que vous aurez apprécier également notre travail !

Notre Travail en PHP

# Outils utilisés

**cURL (libcurl) :** une bibliothèque qui permet de se connecter et de communiquer avec différents types de serveurs, et ce, avec différents types de protocoles. libcurl supporte actuellement les protocoles http, https, ftp, gopher, telnet, DICT, file et LDAP. libcurl supporte également les certificats HTTPS, HTTP POST, HTTP PUT, le téléchargement FTP (ceci pouvant également être effectué via l'extension ftp de PHP), les formulaires de téléchargement HTTP, les serveurs mandataires (proxy), les cookies et l'identification utilisateur/mot de passe.

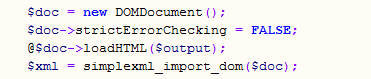
Grace à cURL nous avons récupérer le contenue d'une page (HTTP) afin de l'exploiter par la suite :



**SimpleXMLElement :**

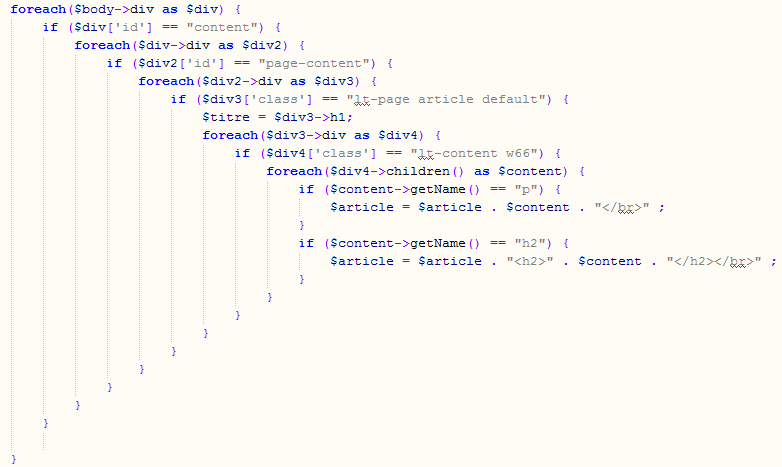
L'extension SimpleXML fournit des outils très simples et faciles à utiliser pour convertir du XML en un objet qui peut être manipulé avec ses propriétés et les itérateurs de tableaux.

Nous avons du convertir le résultat renvoyé par cURL dans un format gérer pour pouvoir utiliser cette extension :

****

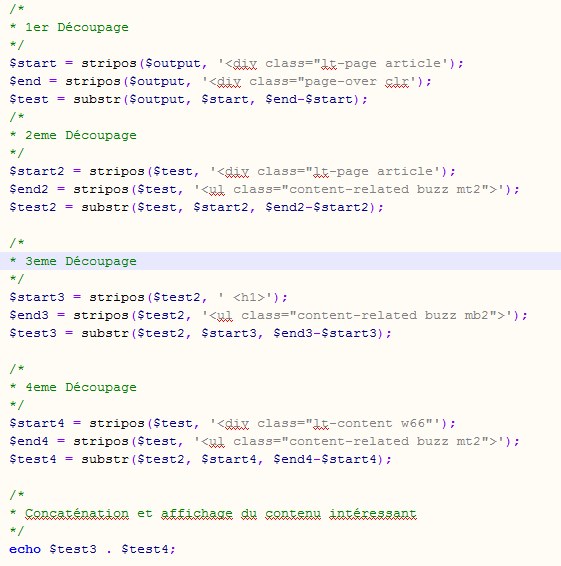
Une fois cette conversion effectuée nous avons pu récupérer uniquement le contenu voulu, c'est à dire le texte d'un article afin de pouvoir le comparer avec les textes des autres articles pour supprimer les redondances.

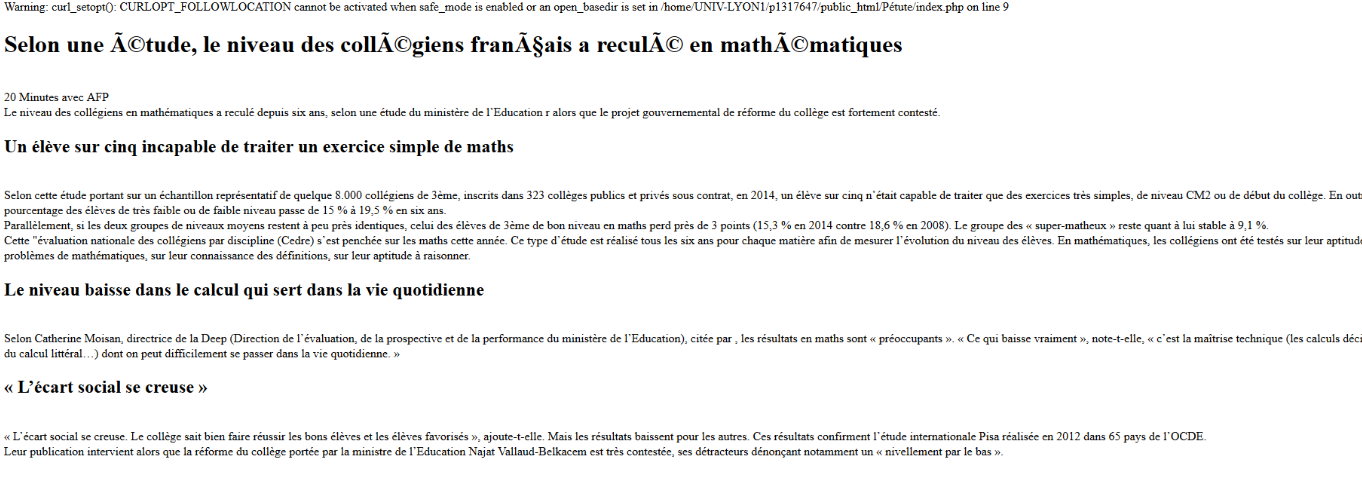
Exemple d'utilisation :



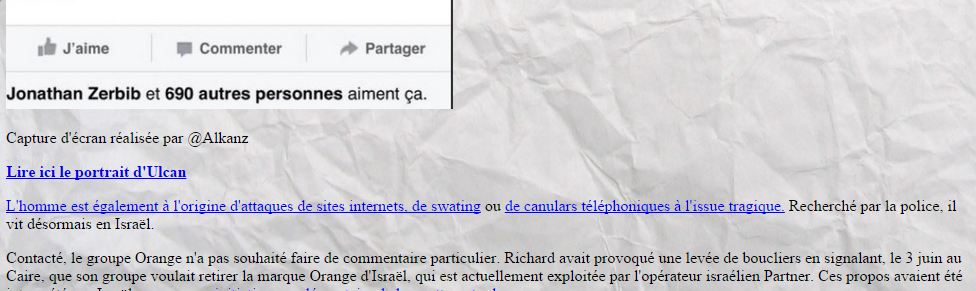
**Algorithme en PHP:**

Afin d'afficher le contenu d'un article il n'est pas nécessaire d'utiliser simpleXMLElement, ceci rend la chose plus complexe et de plus ne conserve pas la mise en page. C'est juste en découpant la chaîne de caractère obtenue par cURL il est possible de récupérer le contenue intéressant (contenu inintéressant | contenu intéressant | contenu inintéressant) .

Exemple :

- Problème des accents dans les titres, images non récupées :

- Après correction du problème en utilisant le second algorithme vu précédemment :



Nous pouvons voir que les urls sont prises en compte, les images également ainsi que les accents dans les titres.

# Difficultés rencontrées

* Problèmes d’encodages avec la récupération des informations, via Curl.
* Fenêtres iframe sont parfois lentes à charger parce qu’elles sont assez lourdes.
* Récupérer toutes les informations contenues dans une page d’actualité et pas uniquement le texte (récupérer images ou urls).