PROGRAMMATION JAVA

Université Lumière Lyon 2

NOUTEPING TATIANA SANCHEZ ALEJANDRA SAQOUT ZINEB PEGDWENDÉ NICOLÁS

Nico

2017

Table Des Matières

[INTRODUCTION 2](#_Toc469320203)

[1. PRESENTATION GENERALE 3](#_Toc469320204)

[1.1. Projet 2 3](#_Toc469320205)

[2. Spécifications 4](#_Toc469320206)

[3. Analyse 5](#_Toc469320207)

[3.1. Quel sera l’environnement de travail (vous pouvez justifier le choix de votre environnement) ? 5](#_Toc469320208)

[3.2. Quelles sont les données identifiées dans les spécifications ? 5](#_Toc469320209)

[3.3. Un diagramme des classes (simplifié bien sûr), c’est-à-dire : 5](#_Toc469320210)

[3.4. Spécifier, par classe, les méthodes publiques identifiées à l’étape précédente : ce qu’elles doivent réaliser (en langage naturel). 5](#_Toc469320211)

[4. Conception 7](#_Toc469320212)

[5. Validation 8](#_Toc469320213)

[6. Codage 9](#_Toc469320214)

[7. Vérification et Maintenance 10](#_Toc469320215)

[8. Conclusion 11](#_Toc469320216)

[9. Perspectives 12](#_Toc469320217)

INTRODUCTION

1. PRESENTATION GENERALE

Dans un monde tourné vers le numerique et marqué par la croissance des réseaux sociaux et du bloging ainsi que par la prolifération des sites web, le volume d’information produit chaque jour est colossale. Ces informations s

Ces données issues des pouvoirs publics, des entreprises, des journalistes mais aussi, et surtout, des utilisateurs eux-mêmes (User Generated Content) représentent

4 milliards d’actions effectuéees chaque jour sur Facebook, 350 000 tweets chaque minute sur Twitter, 72 heures de vidéeo ajoutées chaque minute sur YouTube, 40 millions de nouvelles photos postées quotidiennement sur Instagram, 150 milliards de courriels échangés chaque jour et plus de 800 000 sites créés.

Confronté à ce volume inimaginable de données, majoritairement textuelles, il est diffcile de rechercher une information précise ou même d’avoir un apercu de l’information qui circule sur le Web.

Dans ce contexte, des technologies ont été mises au point pour extraire l’information, la structurer et l’indexer dans des bases de données afin de pouvoir réaliser une analyse efficace de leur contenu (textuel, image, vidéo, etc.). Ce projet est une modeste tentative de développer un système permettant de traiter des données sous un format bien défini, de les indexer et de réaliser des opérations simples de recherche.

1. Spécifications

Nous avons choisi de travailler sur le sujet 2. Ainsi, l’application à développer devra permettre à l’utilisateur d’analyser l’aspect temporel des données. Il devra proposer les fonctionnalités suivantes :

* Permettre l’importation des données à traiter à travers un fichier .csv ;
* Proposer un affichage des données dans un tableau, avec éventuellement une barre de défilement quand la quantité de données est importante ;
* Permettre de trier les articles suivant la date, l’ordre alphabétique des titres ou des auteurs, et/ou la source ;
* Permettre de filtrer les articles par un mot-clef sur le titre, la description, ou l’auteur ;
* Permettre d’exporter le résultat des données filtrées dans un nouveau fichier .csv au même format que le fichier importé ;
* Proposer des statistiques simples d’utilisation des mots sur le corpus filtré (par ex. le nombre d’occurrences des mots) à l’aide de la librairie Lucene, ainsi qu’une évolution de ces statistiques dans le temps.

1. Analyse
   * 1. Environnement de travail

Dans le cadre de ce projet, nous avons choisi de travailler avec l’Environnement de Developpement Integré Netbeans.

**NetBeans** est un environnement de développement intégré gratuit et Open source créé par Sun microsystems (egalement créateur du Java).

Nous avons choisi NetBeans parce qu’il est libre et est l’un des IDE les plus utilisés sur le marché. Ainsi, nous avons une forte communauté d’utilisateurs vers laquelle nous tourner en cas de difficultés. Aussi, Netbeans est simple d’utilisation et permet une prise en main rapide et intuitive.

Diagramme de classes



Description des méthodes *public* par classe:

* Classe ListeArticles
* export() : cette méthode permet l’exportation de la liste dans un fichier .csv
* import() : cette méthode permet l’importation de la liste depuis un fichier .csv
* trier() : Cette méthode permet de trier la liste suivant un champ donné
* Classe ImplementationLucene
* IndexFile() :
* Requete
* Frequence
* Frequences
* ResultatRequete

1. Conception

Pour l’implémentation de l’application, nous avons organisé le travail en tâches élémentaires qui ont été distribuées entre les membres du groupe. Cette distribution a été faite en tenant compte des aptitudes de chaque membre du groupe ainsi qu’il suit.

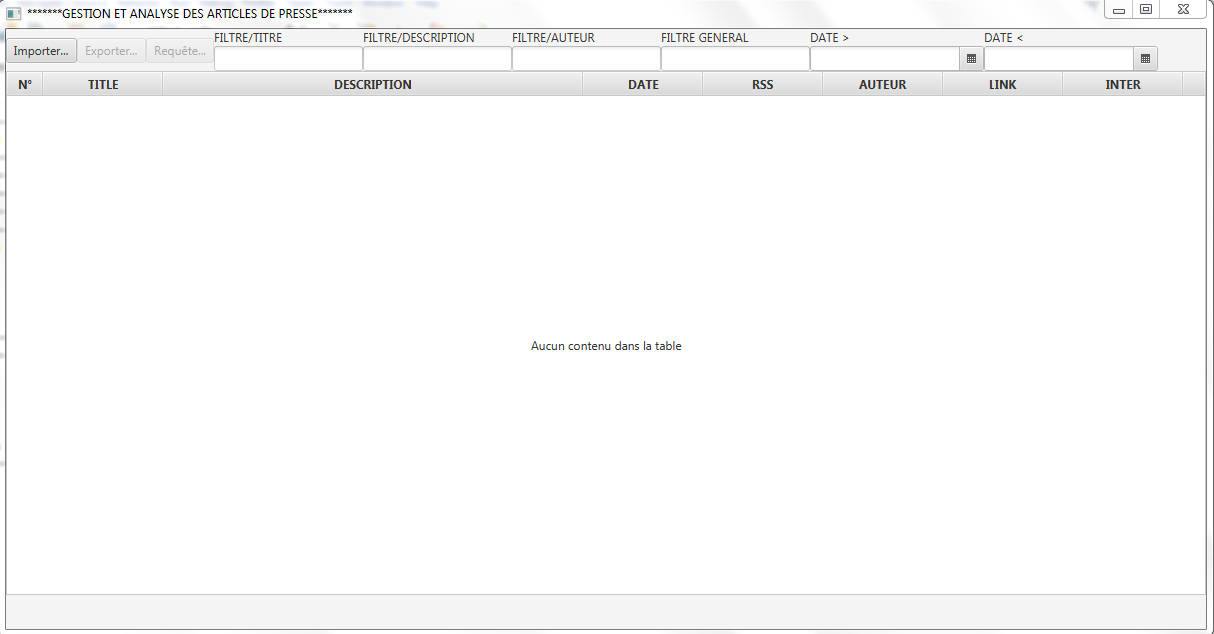
* Alejandra : Creation de la methode d’importation des donnees
* Pegdwendé : Conception de la classe principale et de l’interface graphique et implémentation de la librairie lucene,
* Tatiana : Creation de la methode de conversion des dates
* Zineb : Creation de la methode de conversion d’une ligne CSV en Article, et de la methode de conversion d’un article en ligne CSV.

Au cours de cette phase de conception, nous avons rencontré plusieurs difficultés d’ordres divers. Ce sont principalement :

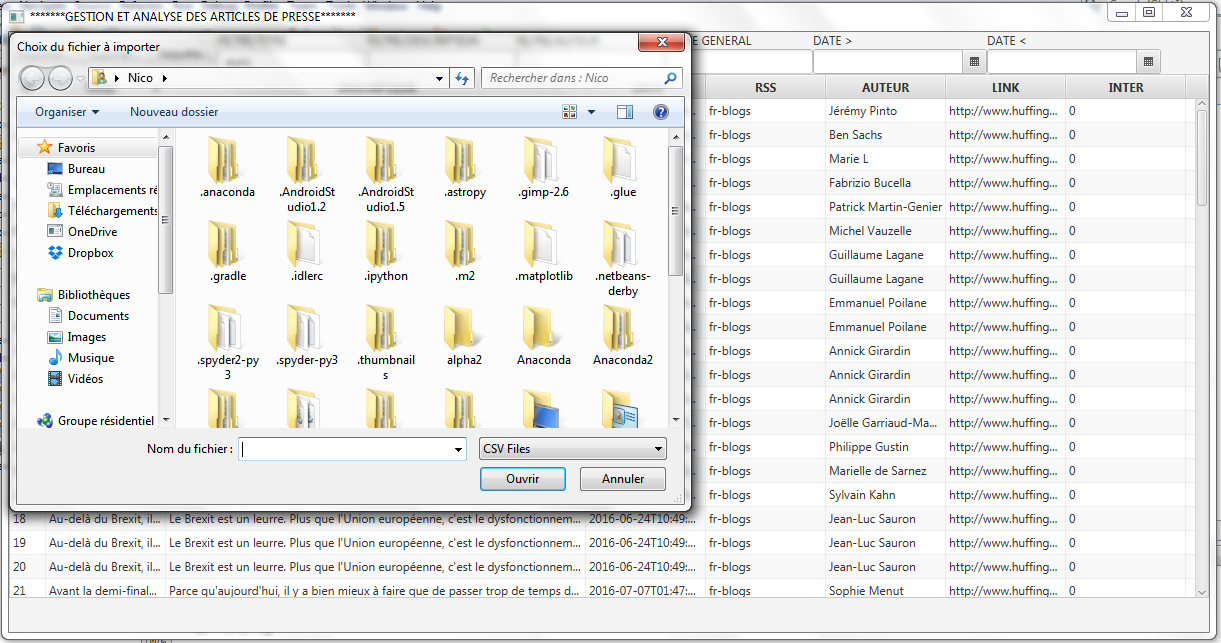
* La gestion de la date : Le format de la date était assez particulier à gerer.
* L

Demo d’utilisation du logiciel

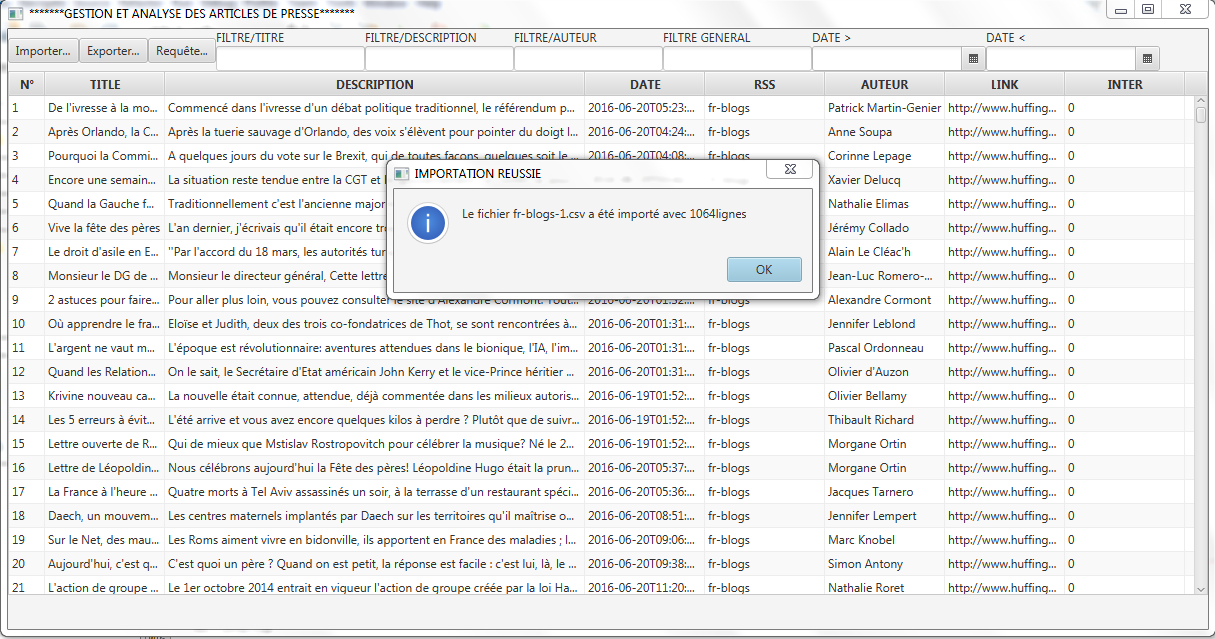
* Au lancement de l’application, la fenêtre s’affiche ainsi



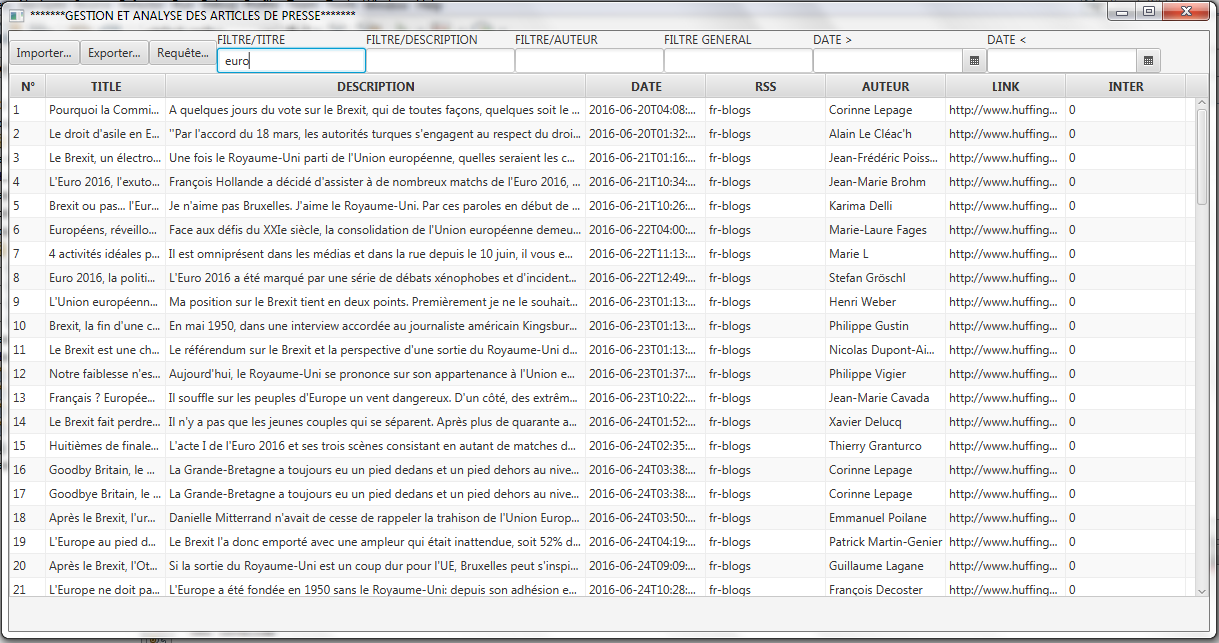
* Lorsque l’utilisateur clique sur le bouton « importer » la fenetre pop-up s’affiche



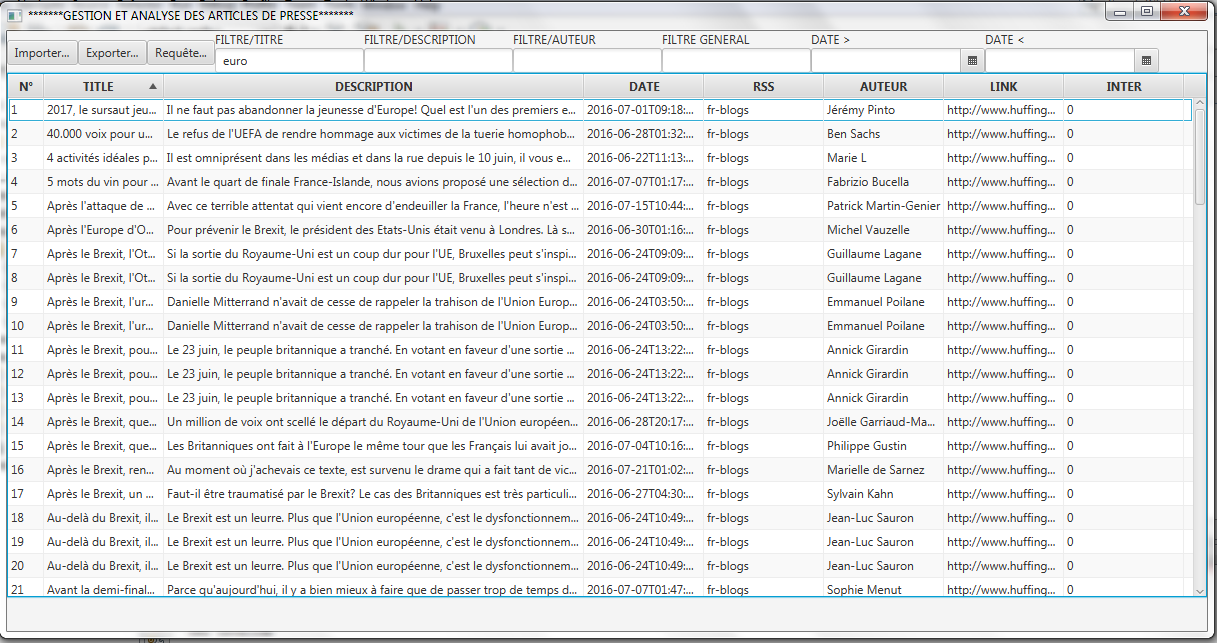
* Une fois le fichier sélectionné, l’importation des données est effectuée



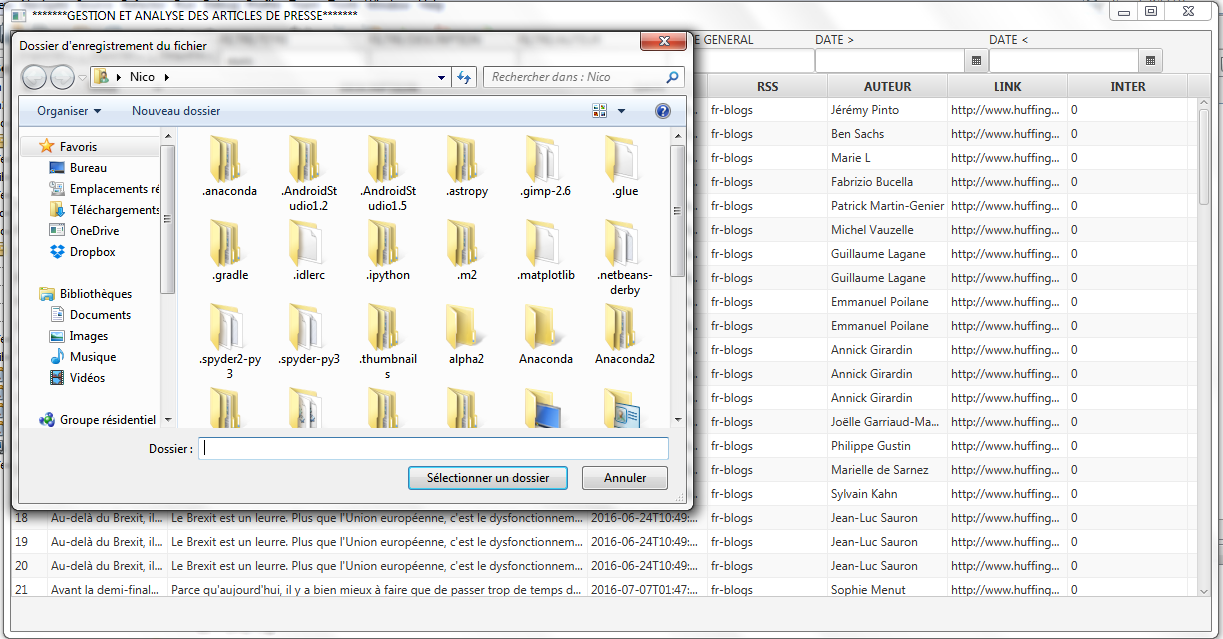
* On peut alors filtrer les données



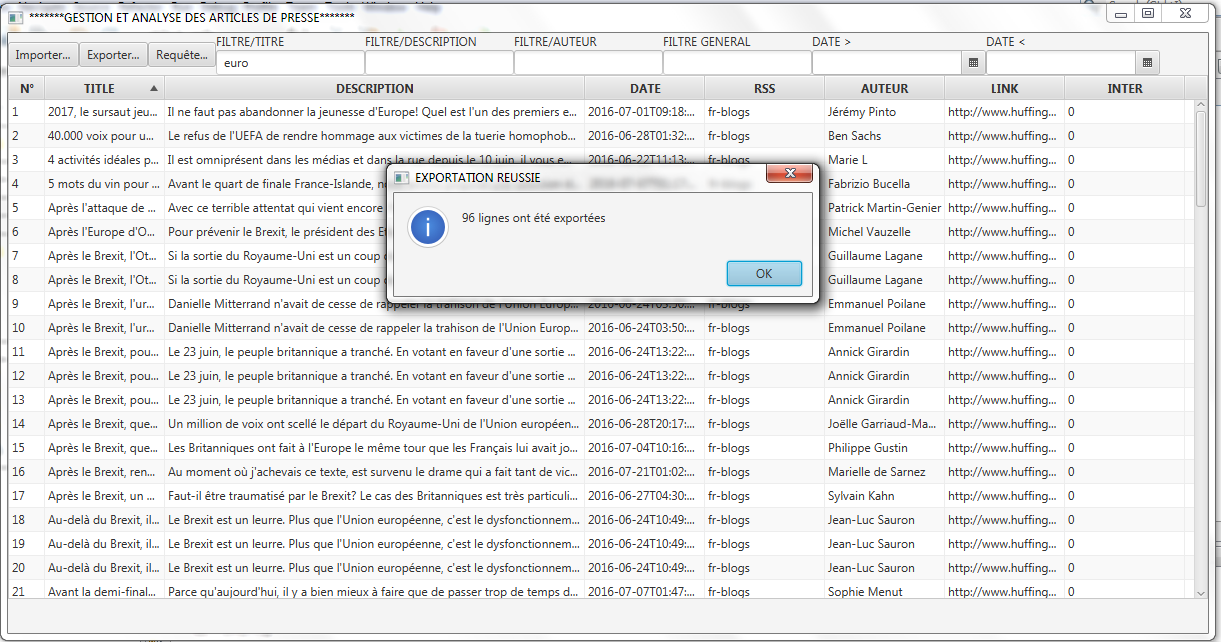
* On peut également appliquer un tri



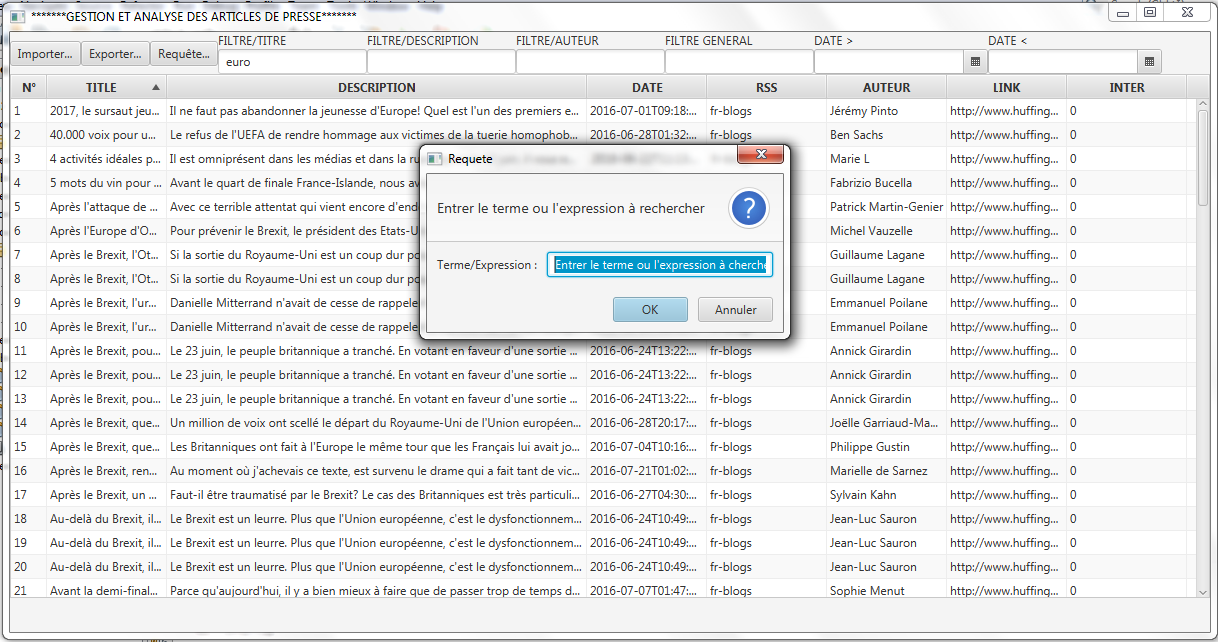
* Pour exporter les données ainsi triées et/ou filtrées, on appui sur « exporter » et une fenetre pop-up s’affiche



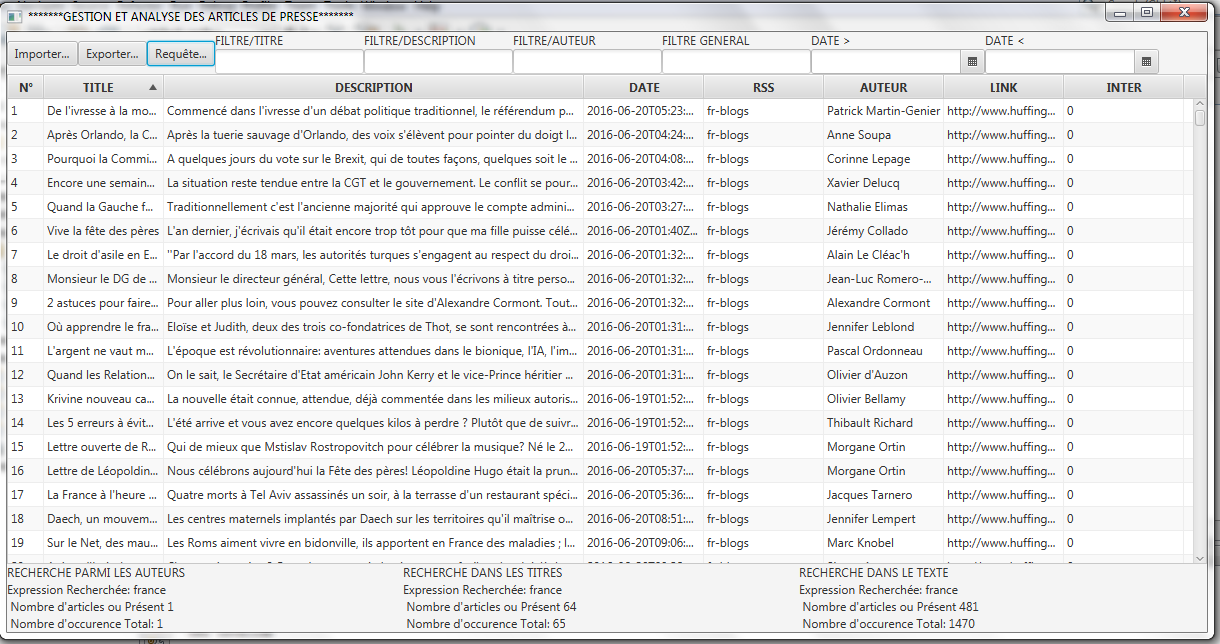
* Une fois le dossier de sauvegarde sélectionné, le fichier est exporté



* Enfin on peut afficher des statistiques en cliquant sur « requete ». Une fenetre pop-up s’affice alors pour recuperer le terme à recercher



* Une fois le terme validé on voit les statistiques sur les occurrences de l’expression dans le texte



1. Validation
2. Maintenance

Une évolution de cette application peut consiter d’une part en une amélioration de l’interface graphique et d’autre part en l’integration de requêtes plus poussée grace à la librairie lucene qui est déjà installée.

Pour ce qui concerne l’interface graphique, on peut envisager l’ajout d’une barre de menu qui rendrait plus intuitive l’utilisation du logiciel. Aussi, l’application pourrait gagner en convivialité à travers l’ajout de contrastes (couleurs, gras) sur la fenêtre. Enfin, la liste ne permettant pas de visualiser l’intégralité des informations sur chaque article, il serait intéressant de pouvoir visualiser un article sélectionné dans une deuxième fenêtre.

Concernant les requêtes avec lucene, une amélioration de ce logiciel peut constituer à l’integration de requêtes permettant d’obtenir automatiquement les mots les plus fréquents et d’en faire un nuage de mots. Aussi, une adaptation automatique de l’index au filtre serait interessant en vue d’harmoniser le resultat des requetes avec les corpus filtré.

1. Conclusion