Auteur :
Joana Celestina D'ALMEIDA ROGER

Encadrant :
Laurent BRISSON

26 juin 2014

Rapport Technique Projet Mineure : Parseur générique de forums

TELECOM Bretagne
Institut Mines-Télécom

Sommaire

1. PREAMBULE

Ce projet s'inscrit dans le cadre de ma formation d'ingénieur généraliste à Télécom Bretagne. « Télécom Bretagne est, à la fois, une grande école généraliste et un centre de recherche international en sciences et technologies de l'information. Elle s'appuie, pour l'ensemble de ses activités, sur un corps professoral permanent de quelque 160 personnes travaillant au sein de 9 départements d'enseignement-recherche ». Pour plus détails http://www.telecom-bretagne.eu/.

2. PRESTATION DU PARSEUR GENERIQUE DES FORUMS

Cet outil utilise la librairie JSoup pour analyser (analyse syntaxique) n'importe quel forum afin d'extraire les informations pertinentes, dans notre cas les commentaires (le contenu du commentaire, l'auteur du commentaire, la date de publication, le lien de la page web sur laquelle se trouve le commentaire), pour ensuite les stocker dans un fichier CSV.

3. GUIDE D'UTILISATION

Dans cette partie nous allons détailler les étapes à suivre pour utiliser ce logiciel.

3.1 IMPORTER UN PROJET DEPUIS GITHUB

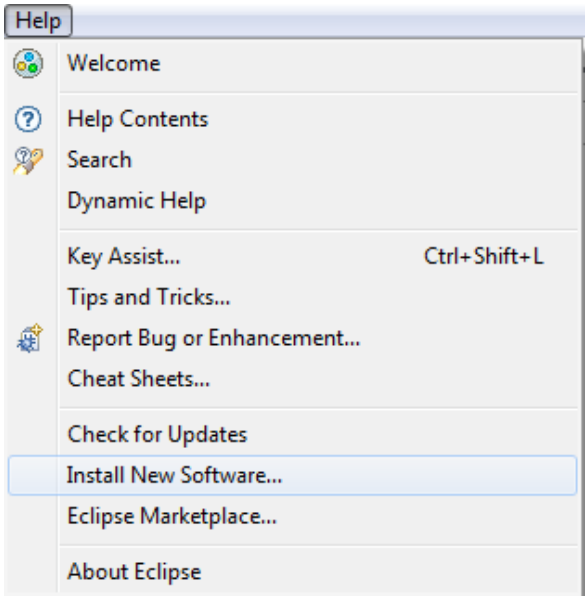
Le code source de notre projet est hébergé sur GitHub, dans cette partie nous allons vous montrer comment importer le projet dans votre environnement de travail afin l'utiliser.
Nous considérons que l'environnement de travail est Eclipse et nous allons vous montrer comment installer le plugin de git dans Eclipse et comment importer un projet depuis GitHub.

3.1.1 Git pour Eclipse

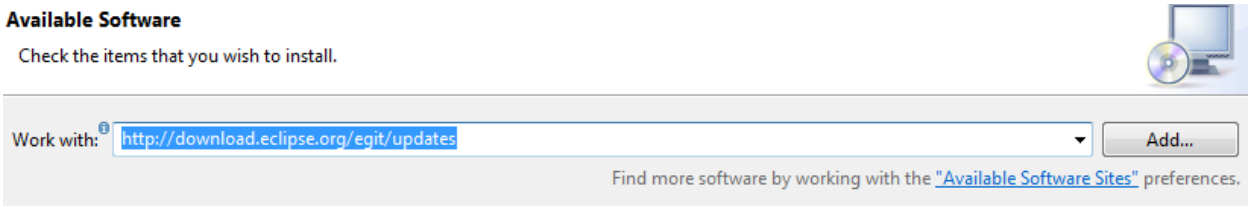
EGit est une extension pour Eclipse qui permet d'utiliser Git. Pour rappel, Git est un logiciel de gestion de versions qui permet à chaque utilisateur d'avoir une copie complète et l'historique d'un dépôt localement et de le distribuer.

3.1.2 Installation

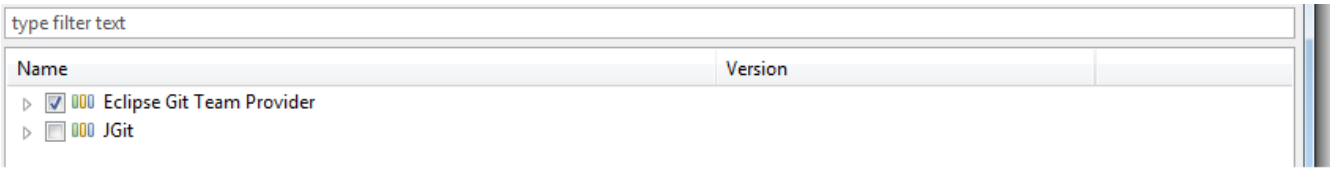
- Pour installer Egit, il faut démarrer Eclipse, cliquer sur le menu Help et ensuite cliquer sur Install New Software ;



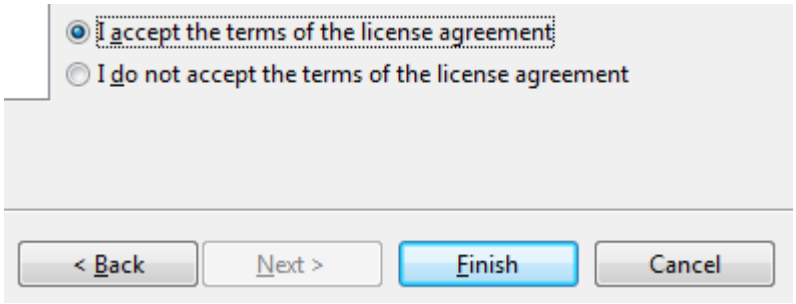
- Il faut préciser l'url pour télécharger le logiciel egit: http://download.eclipse.org/egit/github/updates/;



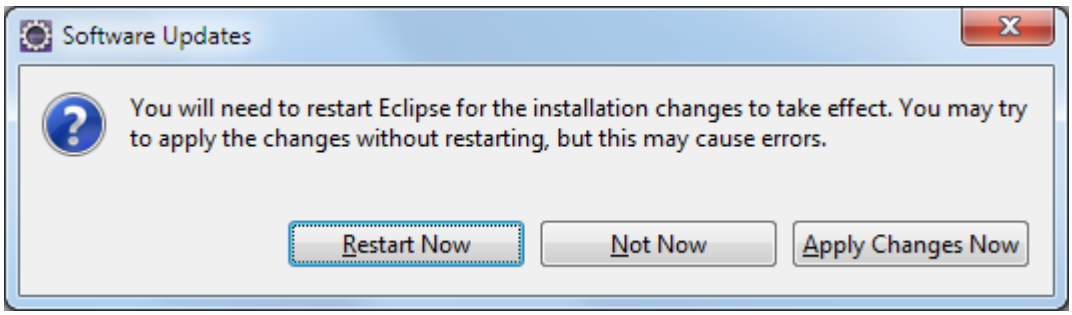
- Ensuite cliquer Enter ;



- Accepter les conditions et cliquer *Finish* ;

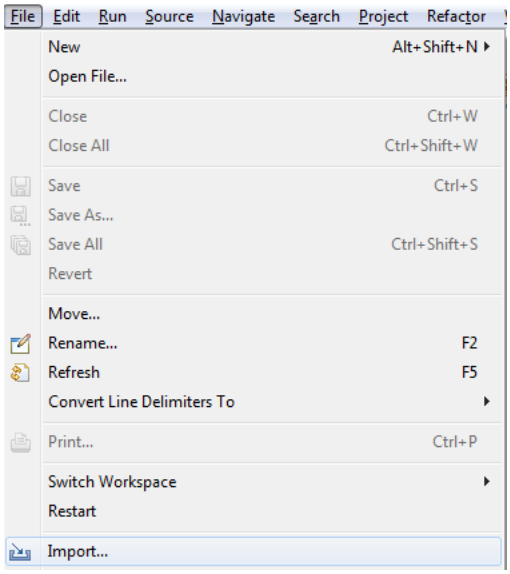


- Redémarrer Eclipse ;



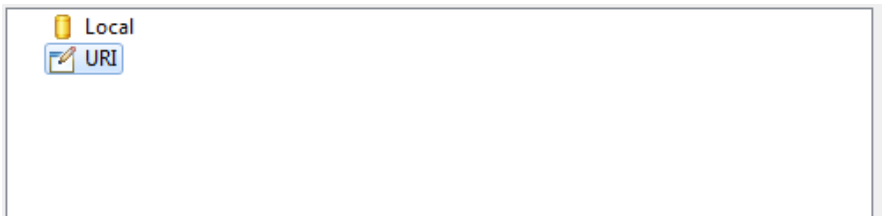
3.1.3 Importer un projet depuis GitHub

- Démarrer Eclipse, cliquer sur le menu File et ensuite sélectionner Import ;

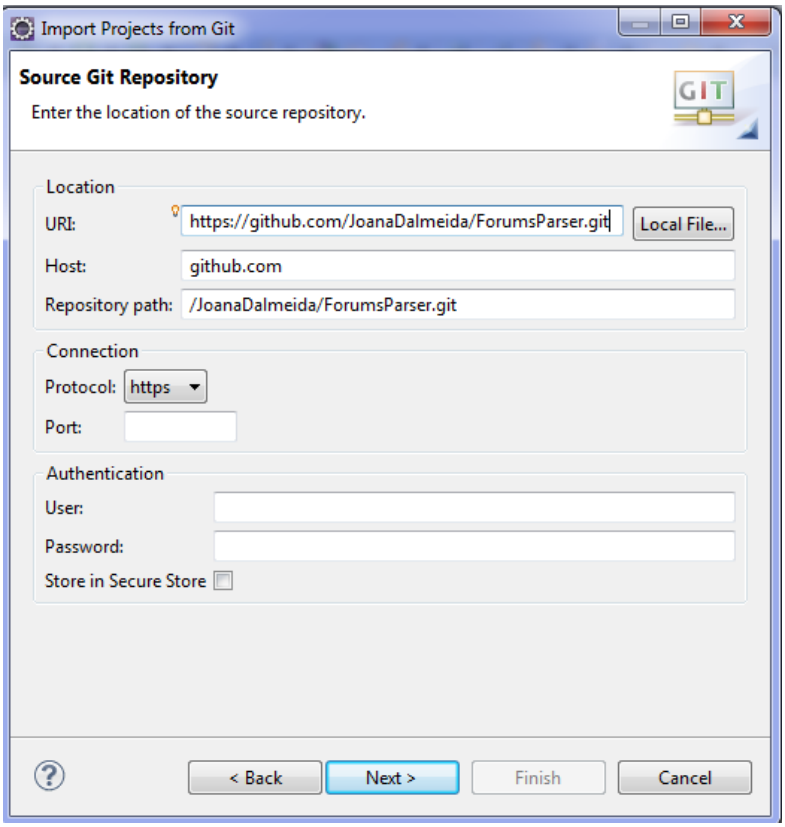


- Cliquer sur le menu Git et sélectionner Projects From Git ;

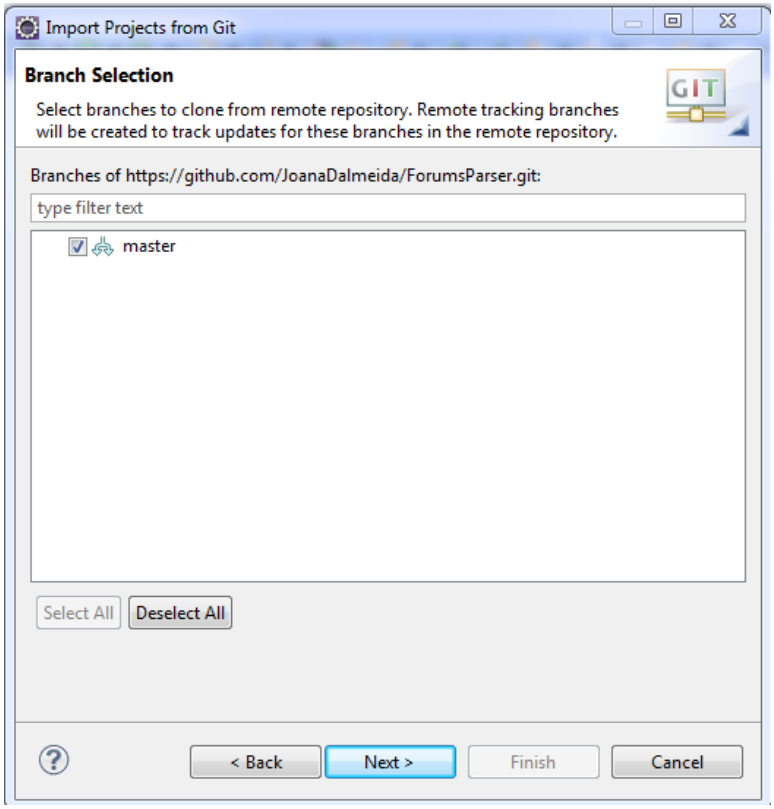
- Ensuite sélectionner URI ;



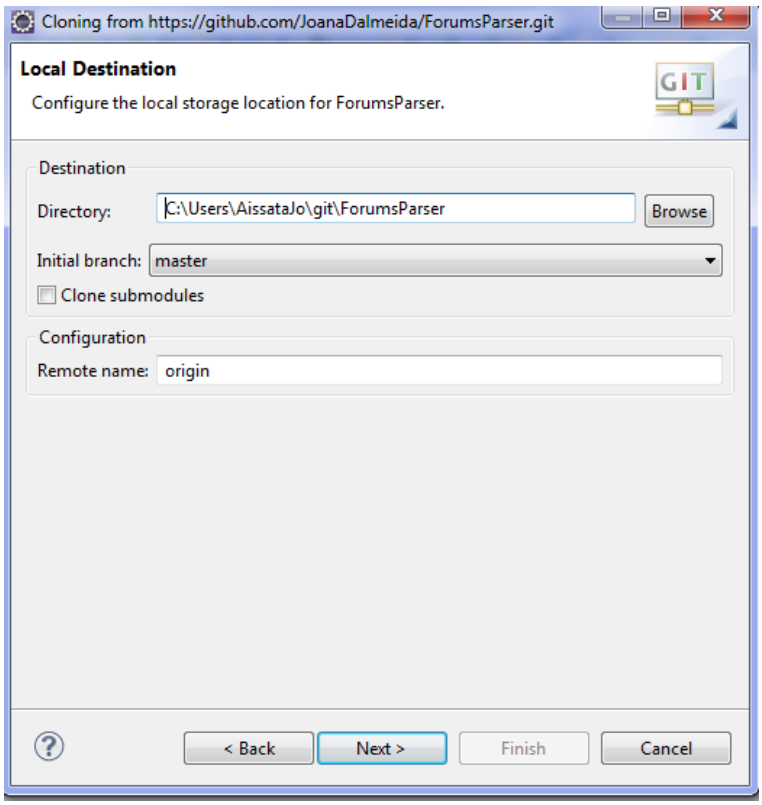
- Il faut se rendre sur la page github du projet https://github.com/JoanaDalmeida/ForumsParser, récupérer l'url du projet et ensuite le coller dans le champ URI ;



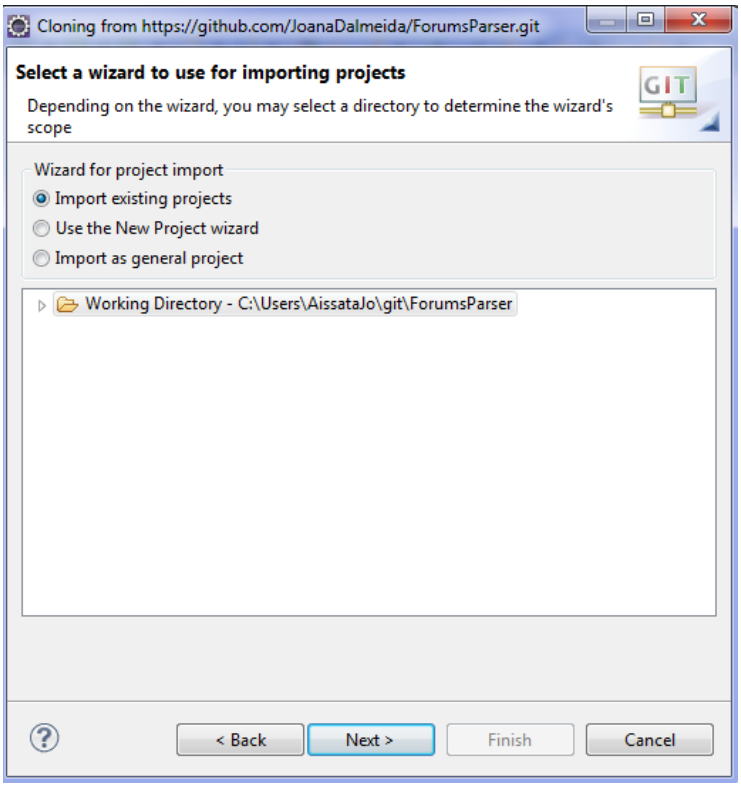
- Ensuite cliquer sur Next ;



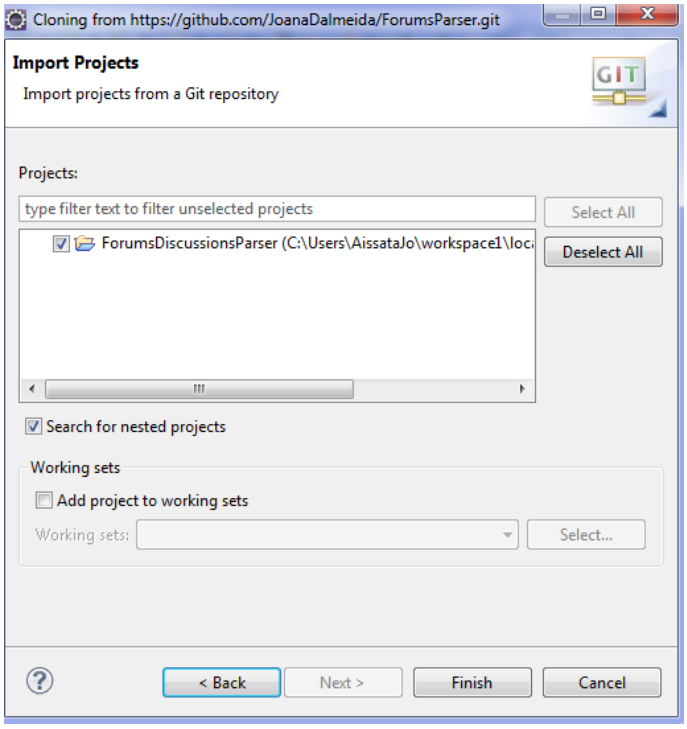
- Cliquer sur Next et sélectionner le répertoire dans lequel vous voulez sauvegarder le dépôt en local ;



- Cliquer sur Next ;



- Ensuite cliquer sur Next ;



Et cliquer sur *Finish* pour finir l'opération.

3.2 PARAMETRAGE

Le programme prend en entrée un fichier de configuration. Ce fichier doit être sous format XML, un exemple de ce fichier (avec des commentaires pour illustrer chaque paramètre) est inclus dans le code source de l'application sur Github. Il contient les inputs qui seront utilisés par le programme pour déterminer les paramètres d'analyse.

Exemple d'input : *<postsFileName></postsFileName>*

La balise « *postsFileName*» doit contenir le chemin du fichier dans lequel seront stockés les commentaires après l'analyse.

3.3 EXECUTION ET RESULTATS

Pour exécuter le programme, il faut accéder à la classe «LaunchWebParsing.java», dans la méthode *main* de cette classe il faut préciser le chemin d'accès au fichier de configuration dans la variable (*String configFilePath = "";)* prévue à cet effet. Ensuite lancer l'exécution de cette classe.

Pendant l'exécution le programme affiche une fenêtre de dialogue afin de faire valider les paramètres d'analyse par l'utilisateur. Le programme génère un ensemble des fichiers à la fin de son exécution. Un fichier dans lequel sont stockés l'ensemble des Urls analysés, un fichier contenant l'ensemble les Urls que le programme n'a pas pu analyser, un fichier contenant les paramètres d'analyse, un fichier pour l'arborescence (forum et sous forum), un fichier contenant les commentaires, un fichier contenant les discussions.

3.4 GESTION DES PAGES NON PARCOURUES

Si le programme n'arrive pas à analyser une page du forum après 3 tentatives, la page est stockée dans un fichier que l'utilisateur a précisé dans le fichier de configuration. L'utilisateur a la possibilité de lancer le programme pour analyser juste ces pages contenu dans un fichier que l'utilisateur doit passer en paramètre dans le fichier de configuration.

Pour cela l'utilisateur doit affecter la valeur « true » au champ *<useUrlsInFile>true </useUrlsInFile>, ensuite* préciser le chemin d'accès au fichier contenant les pages dans la balise *<urlsFileName> </urlsFileName>, préciser* le chemin d'accès au fichier contenant les paramètres d'analyse et affecter la valeur « true » au champ <useParametersFile>true</useParametersFile> qui permet de spécifier au programme d'utiliser le fichier des paramètres d'analyse.

4. SOLUTION TECHNIQUE

Nous avons choisi de développer le programme en Java à cause des avantages que présente ce langage, entre autre la portabilité, il existe plusieurs librairies open sources de parsing en Java.

Pour mettre en place le parseur générique, dans un premier temps nous avons étudié un ensemble des librairies open sources en Java permettant de faire l'analyse syntaxique des sites.

4.1 LIBRAIRIES JAVA POUR PARSER DU HTML

Il existe plusieurs librairies Java permettant de parser les pages HTML. Dans cette partie nous allons présenter quelques librairies les plus connues et les plus utilisées que nous avons étudiées avant d'effectuer notre choix.

Parser	License	Implementation language(s)	Latest date*	HTML Parsing[1]	Clean HTML**	Update HTML***
HtmlCleaner	BSD License[8]	Java	2013-09-05	No	Yes	?
Jaunt API	Jaunt Beta License	Java	2013-08-01	Yes	Yes	No
Jericho HTML Parser	Eclipse Public License	Java	2012-10-30[9]	No??	?	?
jsoup	MIT license	Java	2013-11-18[10]	Yes	Yes	Yes
JTidy	JTidy License	Java	2012-10-09[11]	Yes	?	?
NekoHTML	Apache License 2.0	Java	2013-02-27[13]	No	?	?
TagSoup	Apache License 2.0	Java	2011-07-07	No	?	?
Validator.nu HTML Parser	MIT License	Java	2012-06-05	Yes	?	?

Figure 1 : Liste Librairies Pour Parser HTML. Source : http://en.wikipedia.org/wiki/Comparison_of_HTML_parsers.

4.1.1 HtmlCleaner

HtmlCleaner [1] est une librairie open-source écrit en Java pour parser du contenu HTML. Les contenus HTML des pages web sont généralement désordonnés, mal formé et il est difficile de faire des traitements sur ces contenus. Avant d'utiliser ces contenus, Il est d'abord nécessaire de les nettoyer et mettre les balises, attributs et le texte ordinaire dans l'ordre.

HtmlCleaner réordonne les éléments individuels et produit du XML bien formé pour le contenu HTML d'une page web donnée. Par défaut, il suit des règles similaires que la plupart des navigateurs Web utilisent pour créer le DOM (Document Object Model). Cependant, l'utilisateur peut personnaliser et fournir ses propres règles afin de filtrer certains tags par exemple.

4.1.2 Jaunt API

Jaunt Beta [2] est une librairie libre en Java pour faire l'analyse des pages web. L'API présente un navigateur léger headless (dépourvu de système d'affichage) pour l'interfaçage avec les sites web, les applications web et les services web. Jaunt rend facile l'analyse, le parcours, la recherche du contenu HTML d'une page web et les données XML. Il permet également d'extraire et filtrer facilement les données.

Il dispose de trois niveaux d'abstraction :

- niveau DOM ;
- niveau composant ;

- et niveau navigateur.

Il s'agit d'une API idéale pour l'analyse des pages web où Javascript n'est pas nécessaire.

4.1.3 Jericho HTML Parser

Jericho HTML Parser [3] est une bibliothèque Java permettant l'analyse et la manipulation d'un document HTML, y compris les balises côté serveur, tout en reproduisant textuellement un code HTML non reconnue ou non valide. Il fournit également des fonctions HTML pour faire des manipulations de haut niveau.

4.1.4 Jsoup : Java HTML Parser

Jsoup [4] est une librairie libre en Java pour travailler avec le contenu HTML. Il fournit une API très pratique pour extraire et manipuler des données, en utilisant le meilleur de DOM (Document Object Model), CSS, et les méthodes de Jquery-like.

Jsoup implémente la spécification **WHATWG HTML5[1],** et décompose le HTML en DOM. Ce dernier est le même que celui produit par les navigateurs modernes.

Jsoup permet de :

- parcourir et analyser le contenu HTML à partir d'une URL, un fichier ou une chaîne de caractère ;
- trouver et d'extraire des données, en utilisant le DOM traversal ou les sélecteurs CSS ;
- manipuler les éléments HTML, les attributs et le texte ;
- …

JSoup est conçu pour faire face à toutes les variétés de HTML.

4.2 CHOIX DE LA LIBRAIRIE

Suite à l'étude de quelques librairies, nous avons retenue la librairie libre Jsoup pour notre projet. **Jsoup** présente plusieurs avantages par rapport aux autres librairies.

La facilité d'utilisation de Jsoup constitue une des raisons pour laquelle nous avons choisie la libraire. IL transforme le contenu HTML en DOM, et ensuite il présente un ensemble des fonctions permettant le parcours du DOM, sélectionner les parties qui nous intéressent.

[1] http://www.whatwg.org/

4.3 SOLUTION PROPOSEE

Le logiciel que nous avons développé dans le cadre de notre projet est constitué de deux principaux composants :

- un premier composant dédié au paramétrage du logiciel ;
- un deuxième composant dédié à l'analyse.

Au lancement du programme, nous prenons en entrée un fichier de configuration qui sera lu par le premier composant pour définir les paramètres que le programme, plus précisément le deuxième composant, va utiliser pour analyser le site web voulu.

4.3.1 Paramétrage : Méthode d'analyse de posts

Dans cette phase nous lisons un fichier de configuration sous format xml. Dans ce fichier nous précisons certains inputs que notre programme utilise pour déterminer les paramètres.

Nous demandons à l'utilisateur de nous fournir via ce fichier de configuration un exemple de post. C'est-à-dire nous prenons en entrée un exemple de post (un commentaire, le nom de l'auteur du commentaire, la date de publication du commentaire et la page web sur laquelle se trouve le post), avec ces données nous allons scanner la page web du commentaire, page web passée en input, pour extraire les tags et les attributs (plus précisément la classe CSS) de l'élément qui contient le post.

Nous retenons le tag et les attributs (la classe CSS) de l'élément contenant le post en entier et ensuite nous déterminons le tag et les attributs(la classe CSS) de l'élément qui contient juste le nom de l'auteur, de l'élément qui contient juste la date de publication et de l'élément qui contient juste le commentaire. Après avoir définis les tags et les attributs, nous demandons à l'utilisateur de valider ses informations.

La librairie JSoup nous permet de sélectionner les éléments sur une page web donnée en fonction de leurs tags et leurs attributs.

Maintenant que nous savons comment retrouver nos commentaires sur la page, grâce aux tags et attributs définis, alors nous allons pouvoir parcourir les différentes pages du site web voulu, les analyser et extraire les commentaires.

4.3.2 Gestion de la pagination

Nous avons implémenté deux méthodes pour déterminer les différentes pages à parcourir et à analyser.

- Une première méthode qui consiste à parcourir toutes les pages du site web dans le désordre, nous avons définis des filtres, l'utilisateur peut préciser des mots-clés que doivent contenir les URLs mais aussi les mots-clés d'exclusion c'est-à-dire si un URL contient un des mots-clés d'exclusions, il ne sera pas analysé. La phase d'analyse commence avec l'URL du site voulu (qui est donné dans le fichier de configuration), sur cette page nous cherchons tous les liens qui respectent les critères de mots clés, et on parcourt chaque lien, on l'analyse et on récupère les liens présents sur la page et ainsi de suite. Nous faisons attention à ne pas parcourir deux fois un même lien.

- Une deuxième méthode se base sur la pagination présente sur les pages web, l'utilisateur fournit en entrée ce qui change d'une page à une autre, avec ça nous déterminons les différentes pages pour l'analyse.

4.3.3 Phase d'analyse

Une fois la phase de paramétrage terminée, nous parcourons les différentes pages du site web voulu, pour analyser et extraire les posts (commentaire, l'auteur, la date) et ensuite les stocker dans un fichier CSV.

Lors de l'analyse en cas d'exception de type **Timeout**, nous essayons d'accéder à la même page 3 fois, sans succès nous sauvegardons l'URL dans un fichier donné. L'utilisateur à la possibilité de lancer l'analyse sur un fichier contenant des URLs au lieu d'un site web.

5. LICENCE ET COMMUNAUTE

Cet outil est open source, sous licence Apache 2.0. Le contenu exact de la licence est disponible sur le dépôt du projet sur Github dans le fichier LICENSE à la racine de ce projet. Vous pourrez contribuer au projet sur Github : https://github.com/JoanaDalmeida/ForumsParser.

6. BIBLIOGRAPHIE

[1] HTMLCleaner. Disponible sur http://htmlcleaner.sourceforge.net/ (consulté le 24 mars 2014)

[2] Jaunt API. Disponible sur http://jaunt-api.com/ (consulté le 24 mars 2014).

[3] Jéricho HTML Parser. Disponible sur http://jericho.htmlparser.net/docs/index.html (consulté le 25 mars 2014).

[4] Jsoup. Disponible sur http://jsoup.org/ (consulté le 25 mars 2014).

[5] Egit : Installation. Disponible sur http://www.lennu.net/2012/08/21/egit-configuration-and-creating-a-git-repository-with-eclipse-ide/ (consulté le 10 mai 2014).

[6] Egit : Import From Git. Disponible sur http://www.lennu.net/2012/08/22/import-git-project-into-eclipse/ (consulté le 10 mai 2014).

.

Technopôle Brest-Iroise
CS 83818
29238 Brest Cedex 3
France
+33 (0)2 29 00 11 11
www.telecom-bretagne.eu