



DOCUMENTATION TECHNIQUE ADMINISTRATEUR BOB

Application GNEWS



© ninjava
Télécom Saint Etienne



Table des Matières

1. Introduction.....	2
2. Fonctionnement du système.....	2
2.1. Arborescence générale du projet.....	2
2.2. Les grandes fonctionnalités.....	3
3. Spécificités Techniques	3
3.1. Interactions utilisateur-logiciel.....	3
3.2. Organisation des packages et classes.....	4
4. Fonctionnement de la base de données	12
5. Rôle et actions de l'administrateur	12
6. Les limites du projet	14



I. Introduction

L'application Gnews est une application permettant à l'utilisateur de s'abonner à des flux rss de journaux et donc de lire les articles de ces flux. Mais cette application est également personnalisée pour chaque utilisateur. En effet, celui-ci peut créer un compte utilisateur. Une fois connecté, il s'abonne aux journaux de son choix. Il peut alors « liker » les articles qui lui ont plu, et il sera généré automatiquement un profil utilisateur avec les thèmes favoris de ce dernier. Un journal personnalisé en fonction des likes sera également disponible. Enfin, on affichera une liste d'articles concernant les sujets « hot ».

2. Fonctionnement du système

2.1. Arborescence générale du projet

Le dossier du projet s'articule en **4** dossiers :

- **.settings** : des fichiers nécessaires au fonctionnement d'éclipse.
- **Doc** : il contient la Javadoc du projet.
- **Src** et **bin** : il contient un dossier Images pour l'interface ainsi que nos divers packages et classes implémentées sur Eclipse :
 - o LancementNavigateur,
 - o GestionListe,
 - o HotTopic,
 - o Interface,
 - o Model,
 - o Read.

Notre application fait appel aux logiciels suivant :

- **Eclipse Kepler** utilisant le **JRE System Library JavaSE-1.7** et le **jdk 1.7.0_45** ;
- **WampServer** (SQL, phpmyadmin)
- Pour connecter la base de données et Eclipse :
Un driver **mysql-connector-java-5.1.25-bon.jar**
- Pour l'interface Swing :
forms-1.3.0.jar –WindowBuilder et
miglayout15-swing.jar – WindowBuilder.

2.2. Les grandes fonctionnalités

Voici une liste exhaustive des fonctions du logiciel :

- Collecter des flux de Journaux dans la base de données
- Visualiser des flux de journaux avant toute connection
- Se connecter ou s'identifier sur l'Interface
- Sélectionner ou « liker » un journal
- Parmi les articles proposés, « liker » des articles visualisables dans son compte personnel.
- Sélections des Noms Propres rendus cliquables sur l'interface afin de lancer une recherche du nom sur Wikipédia.
- HotTopic : afficher les articles dont le thème correspond à ceux des articles « likés ».
- Journal personnalisé : les flux proposés à l'utilisateur seront triés et présentés suivant ses likes.

3. Spécificités Techniques

3.1. Interactions utilisateur-logiciel

L'utilisateur pourra interagir avec le logiciel par l'intermédiaire d'une fenêtre de type Swing.

Grâce à cette interface l'utilisateur pourra :

- visualiser les derniers flux de journaux de la base ;
- créer un compte en saisissant login et mot de passe ;
- se connecter en tant qu'administrateur Bob en insérant son Login et son mot de passe ;
- Naviguer ensuite en tant qu'administrateur entre les différentes options de visualisation:
 - o Supprimer les articles de la base
 - o Collecter des articles dans la base
 - o Apercevoir les profils de tous les utilisateurs de l'application
- L'organisation de la table sera insérée dans le dossier du projet et ne nécessitera donc aucune manipulation de la base de données.

⇒ Diagramme de classes :

Le projet contient une partie gestion de l'interface et une partie fonctionnalités de l'application organisée en packages :

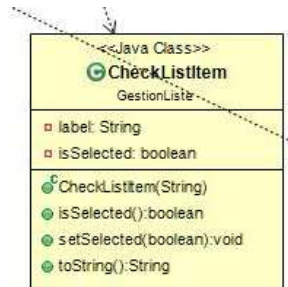
Package GestionListe

Agence la liste des articles dans la fenêtre de l'application avec les cases à cocher.

Ses Classes :

- **CheckListItem.java** :

Classe utilisée pour définir le nom de chaque journal et si celui-ci est sélectionné ou non. (Case cochée ou non)



- **CheckListRenderer.java** :

Création de la liste de journaux avec une case pour chaque journal.



Package HotTopic

Affichage des articles dont le thème s'approche au mieux des articles likés.

Ses Classes :

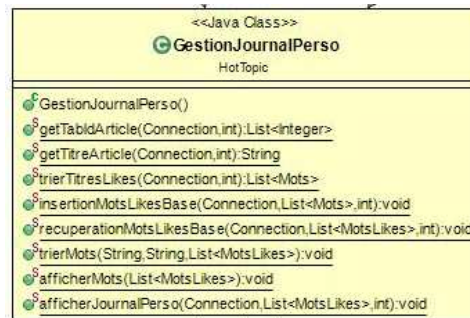
- **GestionHotTopic.java**

Affiche les articles par ordre de plus grande apparition d'un mot liké, c'est-à-dire issu d'un article liké.



- GestionJournalPerso.java

Création du journal personnalisé affichant les abonnements aux flux, les HotTopic, les articles likés par l'utilisateur.



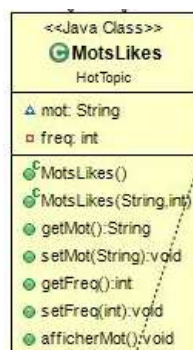
- Mots.java

Récupère les mots des articles sous forme de liste.



- MotsLikes.java

Récupère les mots des articles sous forme de liste afin d'effectuer des calculs de fréquence d'apparition de ce mot dans les autres articles et ainsi trier les flux d'article repéré par leur id.



- TestGestionHotTopic.java

Classe de Test de la récupération et de l'affichage des HotTopics depuis la base de données.



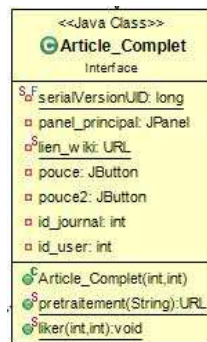
Package Interface

Affichage de l'interface de connexion à l'application.

Ses classes :

- Article_Complet.java

Fenêtre affichant l'article complet avec les noms propres et le bouton pour le like lorsque l'utilisateur est connecté.



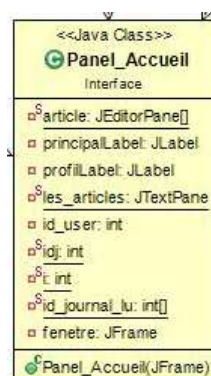
- Fenetre_principale.java

C'est d'ici que la console se lance ! Un clic sur Run...



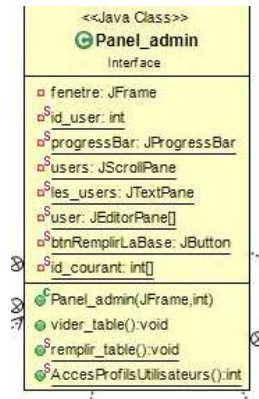
- Panel_Accueil.java

Cette page sera appliquée à la fenêtre lorsqu'on lance l'application. C'est l'affichage de tous les articles et des boutons Créer un compte et Connexion.



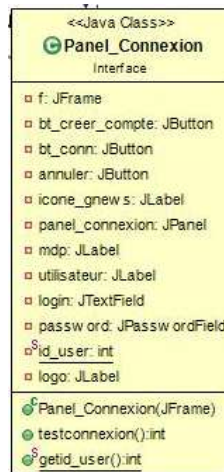
- Panel_admin.java

Cette page est appliquée à la fenêtre lorsque l'administrateur se connecte.



- Panel_Connexion.java

C'est la page qui s'affiche quand on clique sur le bouton Connexion de la page d'accueil. On y entre le login et le mot de passe pour se connecter.



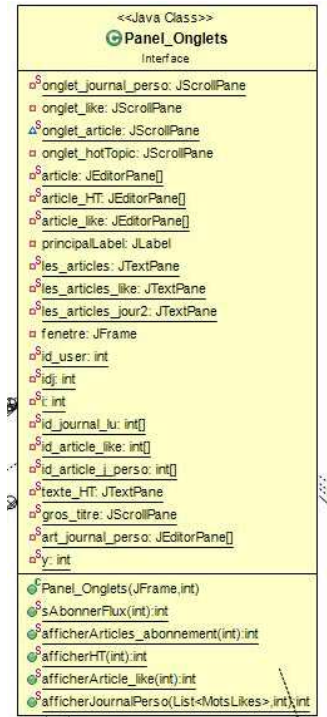
- Panel_CreerCompte.java

Cette page s'affiche quand on clique sur Créer un compte dans la page d'accueil. On y entre le login, le mot de passe et la confirmation du mot de passe. La création est réussie si le login n'est pas déjà utilisé par un autre utilisateur.



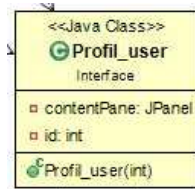
- PanelOnglets.java

C'est la page qui apparait quand l'utilisateur se connecte. Cette page comporte 4 onglets : les articles avec l'abonnement aux journaux, les articles likés, le journal personnalisé, les Hot Topic. Depuis cette page, l'utilisateur peut également accéder à son profil et se déconnecter.



- Profil_user.java

Cette fenêtre s'affiche lorsque l'utilisateur clique sur Mon Profil depuis le Panel_Onglets. On y affiche les journaux auxquels il est abonné ainsi que ses thèmes favoris (à partir des articles qu'il a liké).



- ReadTest.java

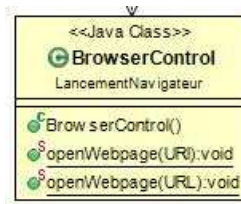
Méthodes qui permettent de vérifier s'il y a des doublons dans la liste des articles. En effet, au chargement des articles dans la base de données, ils ne viennent s'enregistrer qu'une seule fois dans la table, même si ces articles sont encore présents dans la liste des flux.



🔗 Package LancementNavigateur

- BrowserControl.java

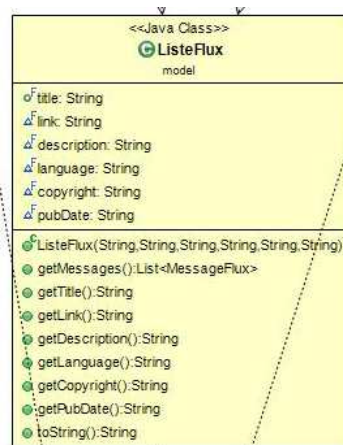
Gère la partie affichage des pages Web sur le navigateur du PC à partir de l'url passée en paramètre.



Package model

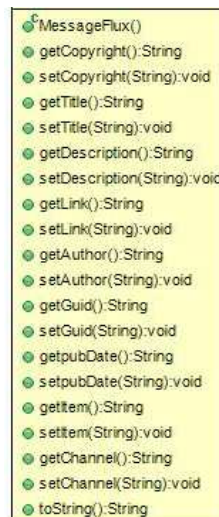
ListeFlux.java

Liste des méthodes permettant de récupérer les informations relatives aux flux RSS. On récupère une liste de flux puis les reconstitue pour les afficher ensuite dans la console.



MessageFlux.java

Un seul Flux est récupéré. Cette classe liste les méthodes permettant de créer le profil utilisateur, de l'afficher et de s'abonner à un flux.



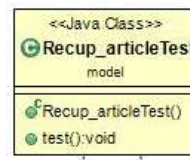
Recup_article.java

La Classe Recup_article récupère le contenu d'un article issu du doc xml, généré par fivefilters.org



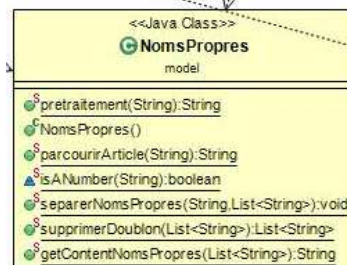
- Recup_articleTest.java

Teste la récupération des articles après uniformisation du doc xml sous fivifilters.org.



- NomsPropres.java

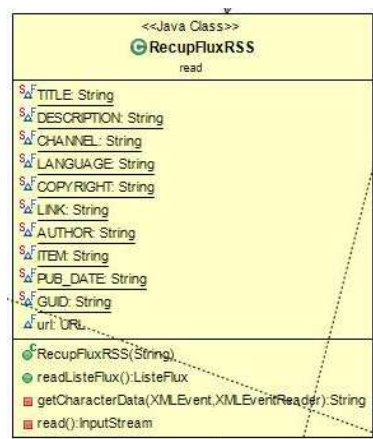
Classe récupérant la liste des noms propres présents dans les articles.



Package read

- RecupFluxRSS.java

Récupérer le flux dans l'url de la page à laquelle on s'abonne. On récupère toutes les informations dans le fichier.



4. Fonctionnement de la base de données


- Table « journal » : table qui contient tous les articles et leurs différents composants et les HotTopics. Quand on lance une recherche de flux, ils viennent s'enregistrer dans cette table.
- Table « prov » : prov = provisoire. On vient enregistrer également tous les articles dans cette table, mais uniquement dans le but de faire une recherche de doublons. Sa structure est identique à la table « journal ».
- Table « utilisateurs » : cette table contient les login, les mots de passe mais également l'abonnement de l'utilisateur, les mots likés et leur fréquence.
- Table « listejournaux » : contient la liste des journaux.
- Table « Likes » : cette table contient les id_journaux dont les articles ont été likés, ainsi que l'id_utilisateur correspondant, nécessaire pour le journal personnalisé.

Pour se connecter à la base : login = « root », mdp = « »

5. Rôle et actions de l'administrateur

Comme expliqué dans la documentation d'installation, l'administrateur doit être le premier à créer un compte sur l'application avec les identifiants de son choix. Il peut alors se connecter en tant qu'administrateur avec ces mêmes identifiants.

La page qui apparaît est la suivante :



Pour remplir la base avec de nouveaux articles

Vous serez informez de la fin du téléchargement grâce à cette barre de chargement.

La collecte peut se révéler assez longue, ne perdez pas patience !

Vous pouvez également vider la base avant de mettre de nouveaux articles

Gnews Bonjour Bob!
Vous êtes connecté en tant qu'administrateur

Déconnexion

Gestion Base de données | Profils des utilisateurs

Collecter les articles

0 %

Cette opération peut prendre plusieurs minutes

Supprimer tous les articles

Attention, toute suppression sera définitive et vous devrez faire une nouvelle collecte pour avoir de nouveaux articles

Le deuxième onglet vous permet de voir la liste de tous les utilisateurs ayant créés un compte sur cette application :



Vous avez alors accès au profil de chaque utilisateur en cliquant sur celui-ci.

La fenêtre suivante vient s'ouvrir :



6. les limites du projet

En suivant bien les étapes d'installation des logiciels du dossier, le projet devrait fonctionner de manière correcte.

Cependant, les fonctionnalités suivantes sont perfectibles :

- **Récupération de l'article** : Les caractères spéciaux contenus dans les articles sont mal affichés dans l'interface. Un post-traitement est appliqué pour les afficher correctement, mais leur liste est non exhaustive et il en restera toujours.
- **Détection des noms-propres** : N'ayant pas trouvé de bibliothèque détectant les noms-propres, nous avons utilisé un système assez basique : la détection des noms propres comme des mots commençant par une majuscule. Certaines limites sont inhérentes à cette méthode :
 - Les mots de début de phrases commençant par une majuscule, nous avons choisi de ne pas les comptabiliser. Si un nom propre est en début de phrase, il ne sera donc pas détecté. Ce problème est atténué par le fait qu'un nom propre apparaît souvent plusieurs fois dans un article, il a donc de bonnes chances d'être détecté à un autre endroit.
 - Si des mots d'un article sont écrits en lettres majuscules ou commencent par une majuscule pour être mis en valeur, ils seront détectés comme des noms propres.
 - Pour les noms-propres composés (comme Grande Bretagne), nous avons mis en place un système qui détecte lorsque plusieurs mots sont à la suite. Cependant si deux noms propres sont à la suite sans être en rapport, ils seront comptabilisés comme un seul nom propre. Exemple : En France Hollande est le président → France_Hollande
 - Les caractères spéciaux liés au premier point peuvent se retrouver dans les noms propres et aboutir à un lien wikipedia incorrect.
- **Hot-Topics** : Les Hot-Topics sont déterminés par analyse de la fréquence des mots dans les titres des articles récents. Les mots les plus fréquents sont les Hot-Topics. On observe par conséquent les limites suivantes :
 - Les mots-blancs : Un certain nombre de mots comme les déterminants ou les chiffres ne sont pas représentatifs d'un sujet. Pour obtenir un résultat cohérent, il faut ne pas en tenir compte. Nous avons bien créé une liste des mots blancs, mais elle est incomplète et il peut arriver qu'un mot blanc devienne malencontreusement un sujet de Hot-Topic.
 - Nous travaillons sur la fréquence de mots seuls. Un meilleur résultat serait obtenu en travaillant sur des groupes de mots, une amélioration possible serait l'utilisation de LDA. Toutefois les Hot-Topics obtenus sont souvent assez proches de la réalité.

- **Journal personnalisé** : Les méthodes utilisées sont proches de celle des Hot-Topics. On a donc les mêmes limites liées aux mots-blancs et au travail sur des mots uniques.
- **Interface graphique** : Les interactions avec l'utilisateur fonctionnent dans l'ensemble très bien. Néanmoins, la classe Panel_Onglet pourrait subir quelques améliorations :
 - Onglet Articles : Lorsque l'utilisateur coche une case, les articles de ce journal s'affichent bien. Mais ils ne s'enlèvent pas s'il décoche la case. De plus, lorsque qu'il se connecte en étant déjà abonné à des journaux, les cases de ceux-ci ne sont pas cochées.
 - Onglet Journal Perso : Le journal personnalisé devrait être affiché comme une vraie Une de journal.