

Projet de Java Feed Calendar

Licence 3 Informatique

forax@univ-mlv.fr, finkel@univ-mlv.fr

But

Le but de ce projet est d'afficher le contenu d'un ensemble de flux RSS/Atom sur des calendriers Google Calendar.

Ce projet est décomposé en deux parties distinctes, la première partie consiste à proposer une architecture de classes et la seconde à écrire le programme respectant l'architecture proposée.

Ce que doit faire le projet

Le projet doit aller lire dans un fichier au format XML (nommé feed-calendar.conf) un ensemble de flux RSS/Atom (des URLs) puis pour chaque flux, créer un calendrier google calendar par flux (si celui-ci n'existe pas déjà) et placé l'ensemble des items du flux sur le calendrier (si ceux-ci n'existe pas déjà) en fonction de leur date de publication. De plus, il devra exister un calendrier général permettant de voir l'ensemble des autres calendriers. Le logiciel devra supporter les flux RSS au format 0.91+, 1.0 et 2.0 ainsi que les flux Atom 0.3 et 1.0.

Première partie du projet

Pour la première partie, vous devez fournir un document contenant les informations suivantes :

- une introduction présentant un peu votre vision du projet
- une étude technique c-a-d une explication de comment installer et utiliser les différentes librairies liées au projets.
L'utilisation de chaque librairie devra être montrée sur plusieurs exemples de codes (vos propres exemples, pas ceux d'un quelconque site Web).
Par exemple, si je veux récupérer les items d'un flux RSS 1.0, je fais cela comment.
- Une description en français de l'architecture de votre projet, en indiquant quels sont les décisions ayant menées à cette architecture (bref le pourquoi) ainsi que l'architecture en elle-même.
La description de l'architecture devra contenir l'ensemble des classes (interfaces, etc.) que vous allez devoir implanter pour votre projet ainsi qu'un diagramme de classes général au format UML.
Pour chaque classe, le document devra contenir une description indiquant :
 - son rôle/sa responsabilité
 - ses relations d'héritage ou d'implémentation avec les autres types
 - les champs de la classes
 - les méthodes de celle-ci

faites en sorte que l'ordre dans lequel vous décrivez les classes soit logiques (donc pas alphabétique)

- Une conclusion

Je vous conseil fortement de faire les tests avant de vous lancez dans la création de l'architecture.

Le format XML du fichier de configuration

Ce format est le même à l'ensemble de vos projet, vous devez donc porter un soin particulier pour en respecter la syntaxe et la sémantique (son sens).

Ce fichier devra contenir les balises :

- **feed-calendar** qui est la balise englobante
- **calendar** qui définit l'accès aux calendriers (Google Calendar)
- **master-calendar** correspond au calendrier à partir duquel tous les autres sont accessibles
- **feeds** qui contient la liste de tous les flux
- **feed** qui décrit un flux avec son nom de calendrier et son URL.

Un exemple complet :

```
<?xml version="1.0"?>
<feed-calendar>
  <calendar login="login@gmail.com" password="password">
    <master-calendar name="All Feeds"/>
  </calendar>
  <feeds>
    <feed calendar-name="planetJDK" url="http://planetjdk.org/feed.atom-1.0">
      <feed calendar-name="planetarium"
url="http://blogs.sun.com/theplanetarium/feed/entries/atom">
        <feed calendar-name="linux" url="http://www.linux.com/feature?theme=rss">
          </feeds>
        </feed-calendar>
```

Conditions de rendu

Le projet est à faire par binome (les mêmes pour la partie 1 et pour la partie 2).

Le document concernant la première partie est à envoyer par mail aux adresses (notez le pluriel) indiquées au début de ce document au plus tard le lundi 15 décembre 23h59 au format PDF.

Références :

Eclipse :

www.eclipse.org

<http://www.forax.org/ens/java-avance/cours/pdf/Eclipse%20pour%20les%20null.pdf>

Accéder à une ressource à partir d'une URL :
les classes `java.net.URI`, `java.net.URL` et `java.net.URLConnection`

Flux RSS :

<http://en.wikipedia.org/wiki/RSS>

Flux Atom :

[http://en.wikipedia.org/wiki/Atom_\(standard\)](http://en.wikipedia.org/wiki/Atom_(standard))

Parseur XML (SAX/DOM) :

<http://java.sun.com/javase/6/docs/technotes/guides/xml/index.html>