



Fil rouge : séance 5

Module FIP INF 211

Gestion de la syndication des informations

Déroulement

À ce stade du projet, les fonctionnalités principales sont implémentées : la manipulation des informations de l'application (consultation/création d'entreprises, d'offres d'emploi, de candidatures) peut être faite via l'interface Web. En plus de l'accès via cette interface, le cahier des charges (pages 12 et 13) prévoit une exportation automatique de certaines des informations (*syndication*) qui pourront être exploitées en dehors de l'application : la liste des offres d'emploi et la liste des candidatures. Les fonctionnalités métier nécessaires à cette exportation existent déjà dans certains des composants développés, il suffira donc de les utiliser.

L'exportation de ces informations se fera en utilisant l'un des formats les plus courants disponible actuellement sur le Web : le *format RSS*. Si le sigle est resté inchangé, sa signification a évolué au cours des différentes versions : dans la version actuelle, 2.0, la signification est *Really Simple Syndication*. Le format RSS est une structuration simple des informations à communiquer qui se base sur XML et est largement utilisé. Un autre format concurrent existe, le format *atom*, mais il ne sera pas abordé ici.

L'objectif de cette séance concerne la manipulation en Java de ce format XML particulier. Elle ne se fera pas en utilisant classiquement un parseur (*SAX*, *DOM*, *JDOM*, etc.) mais avec une approche différente qui utilise le parallèle qui peut être fait entre la structuration des données dans un format XML, exprimée à l'aide d'un schéma, et la structuration en langage Java exprimée à l'aide de classes. La technologie qui supporte cette approche est une technologie livrée en standard avec le JDK : elle se base sur un outil (commande *xjc* pour la génération de classes Java à partir d'un schéma XML) et une API (*JAXB* pour *Java API for XML Binding*).

Pour aller plus loin...

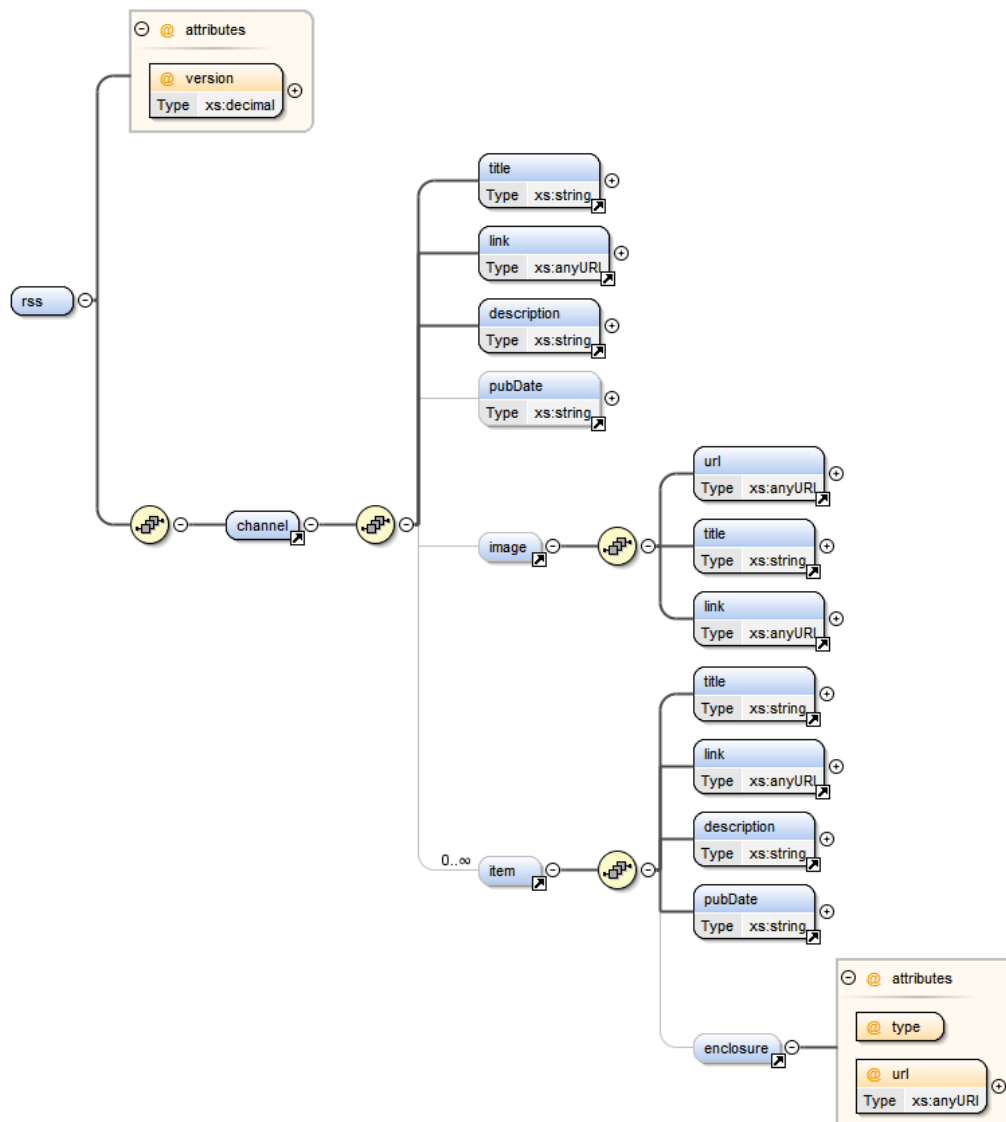
- Exemples de flux RSS :
 - Liste des flux RSS du journal *Le Monde* : <http://www.lemonde.fr/web/rss/0,48-0,1-0,0.html>
 - Liste des flux RSS *Radio-France* : <http://www.radiofrance.fr/boite-a-outils/rss/>
- Informations sur le format RSS (Wikipedia) : <http://fr.wikipedia.org/wiki/RSS>
- Spécifications du format RSS 2.0 : <http://cyber.law.harvard.edu/rss/rss.html>, traduction française : <http://www.scriptol.fr/rss/RSS-2.0.html>
- Présentation de la technologie : <http://www.jmdoudoux.fr/java/dej/chap-jaxb.htm#jaxb-2>
- Tutorial officiel (Oracle) de la technologie JAXB : http://download.oracle.com/docs/cd/E17802_01/webservices/webservices/docs/2.0/tutorial/doc/

Le format RSS

Le format RSS est donc un moyen de consulter rapidement des informations diverses (actualités, réseaux sociaux, etc.) à l'aide de différents outils qui seront les consommateurs de ces flux bruts pour en assurer la visualisation. Différents types d'outils en permettent la lecture :

- Les lecteurs de flux : *Google Reader*, *Feedshow*, *Thunderbird*, *Firefox*, *Internet Explorer*, etc. Certains d'entre eux sont disponibles sous la formes d'outils accessibles en ligne, d'autres sont des outils à installer.
- Les agrégateurs RSS combinent les informations en provenance de plusieurs flux pour les afficher sous la forme de composants graphiques au sein d'une page Web. Parmi les plus connus : *iGoogle* et *Netvibes*.
- Les informations publiées en RSS peuvent aussi être intégrées directement au sein de certains sites Web.

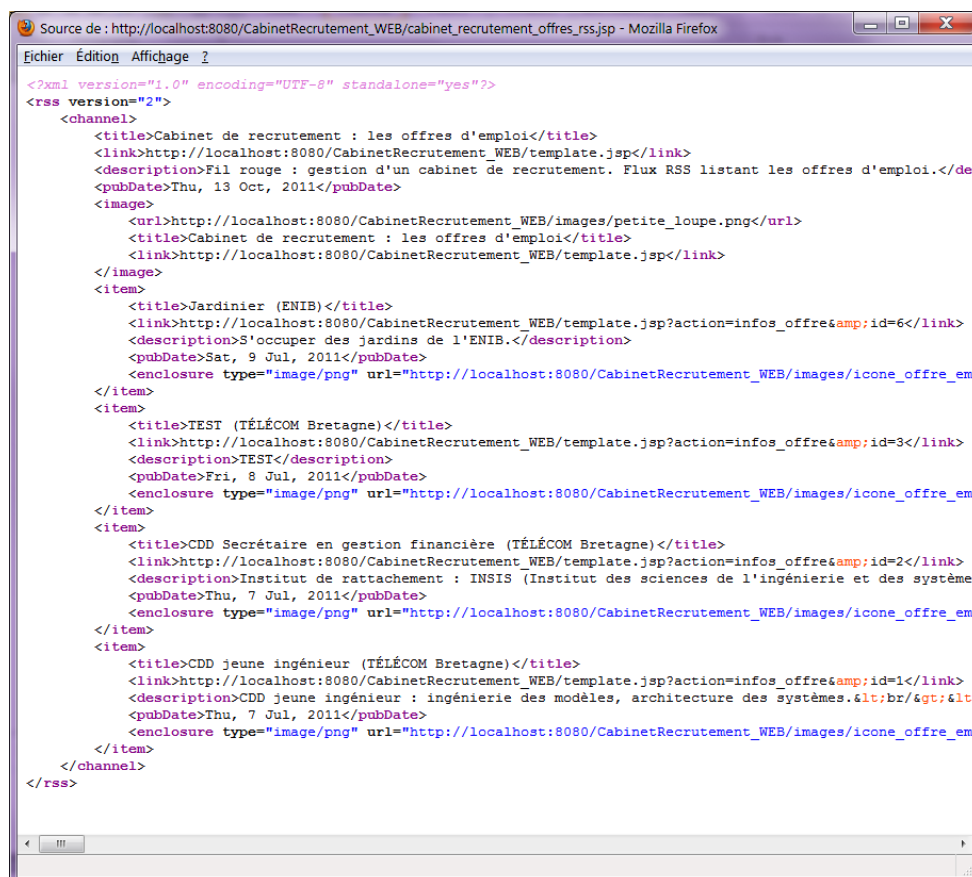
Le format RSS n'est pas standardisé ni normalisé, il existe donc des variations possibles dans le format. Pour la version 2.0, seule une spécification existe (voir le site Web indiqué dans la partie *Pour aller plus loin...*). Pour utiliser JAXB, il est nécessaire de partir d'un schéma XML représentant le format afin de générer les classes. Celui utilisé dans l'application a été généré à partir d'un exemple de flux RSS avec l'outil *Oxygen* : il a été simplifié mais reste totalement compatible avec les lecteurs et agrégateurs RSS. La représentation graphique de ce schéma XML (disponible sur Moodle) montre la structure d'un flux RSS. Des commentaires provenant de la spécification RSS explique la structure du format RSS sont disponibles au sein du fichier.



Représentation graphique du schéma XML décrivant le format RSS



Visualisation du flux dans l'agrégateur Web Netvibes



Exemple de sérialisation XML du flux RSS des offres d'emploi

La racine du document XML est l'élément *rss*, il contient un unique élément : *channel*, qui représente la réelle racine du flux. Cet élément contient trois sous-éléments obligatoires :

- *title* : le titre du flux. Ici, *Cabinet de recrutement : les offres d'emploi*.
- *link* : le lien vers la racine du site. Dans notre cas, la page d'accueil de l'application cabinet de recrutement.
- *description* : une courte description du flux. Cette description n'est pas toujours utilisée dans les visualisateurs de flux mais elle est néanmoins obligatoire dans la spécification.

Les autres sous-éléments sont optionnels :

- *pubDate* : date de la dernière mise à jour du flux.
- *image* : image représentant le flux. Même présente, elle n'est pas toujours utilisée (*Netvibes* ne s'en sert pas mais *Internet Explorer* la visualise). Si elle est présente, les trois sous-éléments qui la constituent sont obligatoires : *url*, *title* et *link*.
- Une suite d'éléments *item*. Cette suite peut être vide (aucun élément *item* : dans ce cas, le flux n'est pas justifié !). Chaque *item* représente une information. Dans l'application, l'information élémentaire est une offre d'emploi. L'information est, elle-même, structurée dans plusieurs sous-éléments, tous obligatoires sauf l'élément *enclosure* :
 - *title* : le titre de l'offre.
 - *link* : le lien qui permettra d'afficher les informations de l'offre (un JSP, *infos_offre.jsp* par exemple renseigné avec l'id de l'offre à afficher).
 - *description* : le description de la mission.
 - *pubDate* : la date de dépôt de l'offre.
 - *enclosure* : élément optionnel, permet de lier un objet média à l'item. Dans l'exemple, on l'associe à une image (attribut *type* égal *image/png*) et on indique dans l'attribut *url* le lien pour l'afficher. C'est cette image qui est affichée par *Netvibes* pour chacune des offres.

Java API for XML Binding (JAXB)

Les données constitutives du flux sont structurées dans les éléments XML définis dans le schéma. Il aurait été possible de le générer en utilisant classiquement un parseur XML qui aurait récupéré les données auprès d'un des composants métier de l'application pour construire à la volée le flux XML. L'approche utilisée dans cette séance est différente, elle pourrait être comparée (grossièrement...) à la technologie du mapping objet/relationnel où on fait une équivalence entre les tables/colonnes de la base de données sous-jacente et les objets/attributs qui sont manipulés dans l'application. Dans le cas de la technologie JAXB, il est possible de faire une équivalence entre les éléments XML définis dans le schéma et les classes du monde Java.

L'API JAXB est une API standard disponible par défaut dans le JDK. Les classes qui la composent font partie du package *javax.xml.bind*.

Les fonctionnalités de cette API sont doubles :

1. Elle permet de générer un flux XML (ou un fichier, etc.) à partir d'instances de classe : fonctionnalité de **sérialisation** appelée en anglais *marshalling*. Cette fonctionnalité est celle qui sera utilisée dans l'application.
2. Elle permet aussi la fonctionnalité inverse, créer des instances de classes à partir des données en provenance d'un flux XML : fonctionnalité de **désérialisation** appelée en anglais *unmarshalling*. Cette fonctionnalité ne sera pas utilisée dans l'application : elle produit du XML mais n'en consomme pas.

Via cette API, la représentation des données dans l'application (en Java donc) se fera sous la forme d'instances de classe. La première étape consiste donc à générer ces classes.

Associé à l'API JAXB, un outil est disponible en ligne de commandes : *xjc*. Cet outil est un compilateur qui, à partir d'un schéma XML, génère le code source des classes correspondant aux éléments XML du schéma.

Travail à réaliser

Génération des classes

Dans notre cas, à partir du schéma représentant le format RSS seront générées les classes suivantes : *Rss*, *Channel*, *Item*, *Image* et *Enclosure*.

Pour générer les classes :

1. Récupérer sur Moodle l'archive ZIP *RSS.zip*. Un fois décompactée, le schéma XML se trouve dans le fichier *RSS.xsd*.
2. L'utilisation de cet outil se fait au sein d'un terminal. L'option `-p` sera utilisée pour fixer le nom du package des classes générées :

```
xjc -p eu.telecom_bretagne.cabinet_recrutement.front.rss RSS.xsd
```

Le choix du package est laissé à votre appréciation. Le choix fait ici place la gestion du flux RSS dans l'interface graphique (package *eu.telecom_bretagne.cabinet_recrutement.front*) où un sous-package qui lui est dédié est créé : package *rss*.

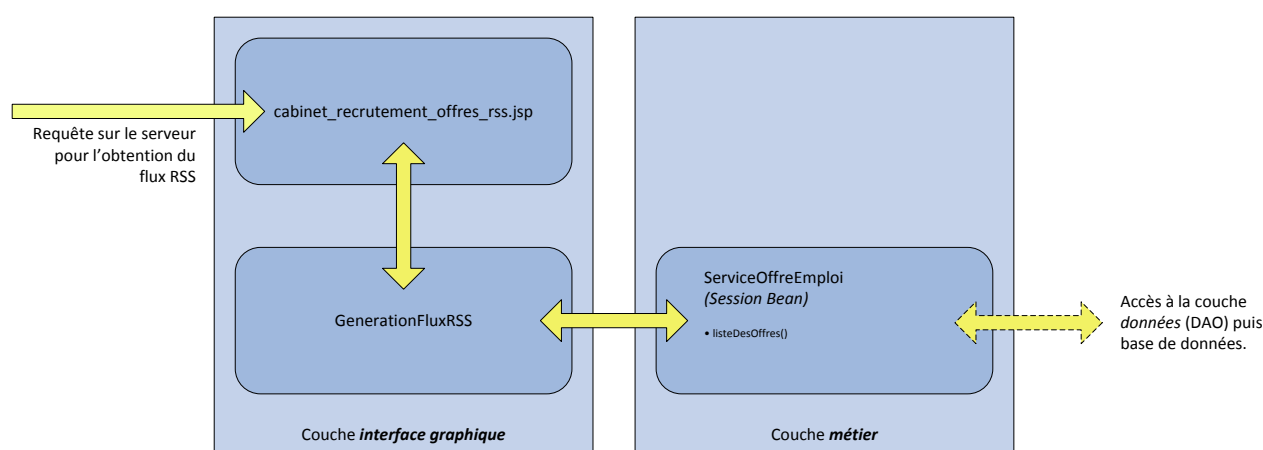
xjc génère alors les fichiers suivants :

- *eu/telecom_bretagne/cabinet_recrutement/front/rss/Channel.java*
- *eu/telecom_bretagne/cabinet_recrutement/front/rss/Enclosure.java*
- *eu/telecom_bretagne/cabinet_recrutement/front/rss/Image.java*
- *eu/telecom_bretagne/cabinet_recrutement/front/rss/Item.java*
- *eu/telecom_bretagne/cabinet_recrutement/front/rss/Rss.java*
- *eu/telecom_bretagne/cabinet_recrutement/front/rss/ObjectFactory.java*

Outre les cinq classes structurant les données dans le flux RSS, une sixième classe est générée : *ObjectFactory*. Cette classe a un rôle utilitaire : elle contient un ensemble de méthodes (*create**) permettant de créer les instances : *createRss()*, *createItem()*, etc.

3. Intégrez ces classes dans le répertoire *src* du projet *CabinetRecrutement_WEB*.
4. Examinez le contenu de ces classes pour en comprendre la structure. Cette compréhension est nécessaire pour la suite du travail.

Génération du flux RSS



Principe de la génération du flux RSS

Le flux RSS s'obtient en envoyant une requête (HTTP) sur un JSP déployé dans l'application Web : *cabinet_recrutement_offres_rss.jsp*. Ce fichier est disponible dans l'archive *RSS.zip*. La seule action du JSP est d'appeler

la méthode *offresEmploi(...)* de la classe *GenerationFluxRSS* dont l'ébauche est aussi disponible dans l'archive. Cette classe récupère les données par l'appel du composant EJB *ServiceOffreEmploi* (dont le nom peut différer dans votre projet). La logique métier existe et est déjà utilisée dans l'application. La fonctionnalité de génération du flux RSS ne fait que de l'utiliser dans un contexte différent.

Le chaînage des différents appels est déjà réalisé, une fois les deux fichiers intégrés dans l'application, le travail se déroulera au sein de la classe *GenerationFluxRSS*.

1. Préparation de l'environnement de travail :

- Recopiez le JSP *cabinet_recrutement_offres_rss.jsp* dans le répertoire *WebContent* du projet *CabinetRecrutement_WEB*.
À noter : il n'y a pas de passages à la ligne dans ce JSP : NE PAS EN AJOUTER ! Cela perturberait la génération du flux.
- Recopiez la classe *GenerationFluxRSS* dans le même package que les classes générées par *xjc* : package *eu.telecom_bretagne.cabinet_recrutement.front.rss*.

2. Dans la méthode *offresEmploi(...)*, adaptez éventuellement la récupération du composant EJB permettant de récupérer la liste des offres d'emploi.

À ce stade, un flux RSS minimal ne contenant aucune information peut être récupéré. Ce flux n'est pas conforme au schéma mais est correctement détecté par le navigateur Web comme étant un flux RSS. L'accès au flux se fait via l'URL : *http://localhost:8080/CabinetRecrutement_WEB/cabinet_recrutement_offres_rss.jsp*. Le source XML de son contenu est le suivant :

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rss version="2">
  <channel/>
</rss>
```

3. Complétez cette méthode: voir les commentaires de la classe pour la réalisation de ce travail.

Prendre en compte dans un premier temps les éléments obligatoires du format RSS et les éléments optionnels dans un second temps.

Remarques :

- Pour ces images, il est possible d'utiliser celles présentes dans le répertoire *WebContent/images* du projet *CabinetRecrutement_WEB*.
- Pour la gestion des dates, la classe *Utils* (package *eu.telecom_bretagne.cabinet_recrutement.front.utils*) offre une méthode gérant le format de date exigé dans la spécification RSS : méthode *date2StringRSS(Date uneDate)*.

4. Une seconde méthode, *candidatures*, est présente dans la classe. Elle permet de récupérer un flux RSS contenant la liste des candidatures.

La visualisation de ce flux nécessite un JSP spécifique qui n'est pas présent qu'il faut donc créer.