
TP 8

Les objectifs de ce huitième TP sont :

- l'écriture de parseurs avec XDOM en lecture et en écriture.

Ce TP est l'examen que j'ai donné le 13 décembre 2012 aux étudiants de GI. L'énoncé a été légèrement modifié car il autorisait l'étudiant à choisir son type de parseur : XmlReader/XmlWriter, DOM ou XDOM. Dans le cadre de ce TP, nous imposons XDOM.

N'oubliez pas que pour vous entraîner, vous trouverez sur l'ENT les annales des années passées.

Sujet XML Parseurs – flux RSS

CONTEXTE

RSS (Really Simple Syndication) est une famille de formats XML utilisés pour la syndication de contenu Web. Un flux RSS est surtout utilisé pour la publication fréquente (blogs, sites, d'informations, audio, video, podcasts...) dans un format standard. Un document RSS (dit «web feed» ou «channel») définit pour chaque nouvelle information, une description résumée et des méta-informations (dates, liens, auteurs...).

Les lecteurs peuvent s'abonner via des agrégateurs (web-based, mobile-based, desktop-based...).

RSS est donc utilisé pour faire du «fast browsing» de news et de mises à jour. Sans RSS, les utilisateurs devraient vérifier votre site chaque jour pour vérifier s'il a été actualisé. Grâce à RSS, ils peuvent vérifier les mises à jour de plusieurs sites en un seul affichage.

Les données véhiculées par RSS sont simples et légères et peuvent donc être chargées par des téléphones portables.

Un fichier RSS est un document XML avec une extension .xml ou .rss. Il contient un élément racine rss, contenant lui-même un seul sous-élément channel. Ce dernier contient les éléments title, description, link, puis plusieurs éléments item. Chaque item correspond à une «news» et contient lui-même des éléments title, description, link, pubDate...

```

<?xml version='1.0' encoding='UTF-8' ?>
<rss version="2.0">
  <channel>
    <title>Cours d'XML</title>
    <description>cours pour les étudiants de GI</description>
    <link>http://.../XML_GI_2009_2010.html</link>
    <item>
      <title>Exemple RSS</title>
      <description>un exemple de flux RSS</description>
      <link>http://.../XML_GI_2009_2010.html</link>
      <pubDate>Sun, 29 Nov 2009 01:52:00 +0100</pubDate>
    </item>
    <item>
      <title>Mise à jour du cours</title>
      <description>cours sur les schémas XML</description>
      <link>http://.../XML_GI_2009_2010.html</link>
      <pubDate>Sun, 15 Nov 2009 00:15:00 +0100</pubDate>
    </item>
  </channel>
</rss>

```

prologue XML

déclaration de document RSS
le flux (métadonnées - unique)

titre du flux
description du flux
URL du site correspondant au flux

un article

son titre
description succincte
URL de l'actualité
date de publication

un autre article

OBJECTIFS

Le but de l'exercice est de parseur des flux RSS du site des émissions de France Inter, aussi bien en lecture qu'en écriture. Vous devrez donc dans un premier temps, afficher le contenu d'un flux RSS dans un format donné, puis dans un deuxième temps, permettre l'ajout d'un item au flux RSS. Vous devrez réaliser cet exercice en C# en utilisant XDOM.

Attention, vous remarquerez en ouvrant les fichiers que France Inter (ils ne sont pas les seuls) utilise dans ses fichiers RSS des balises supplémentaires provenant de deux autres espaces de noms : "<http://www.itunes.com/dtds/podcast-1.0.dtd>" et "<http://radiofrance.fr/Lancelot/Podcast#>", pour la gestion des podcasts sous différents logiciels.

CE QUI VOUS EST FOURNI...

Pour vous permettre de concentrer vos efforts sur les parseurs, différentes classes vous sont déjà fournies :

- **XMLTags** est une classe statique contenant des chaînes de caractères constantes correspondant aux différents éléments, attributs, espaces de noms, préfixes... des documents XML fournis.
- **Item** contient un titre (**Title**), une description (**Description**), une date de publication (**PubDate**), un URL (**EnclosureURL**), un auteur (**Author**) et une méthode **ToString** s'occupant de transformer une instance dans le format demandé.
- **Channel** contient un titre (**Title**), une description (**Description**) et une collection d'items (**Items**), ainsi qu'une méthode **ToString**.

Vous pouvez utiliser, modifier et maltraiter ces trois classes dans votre projet.

De plus, 6 fichiers RSS vous sont fournis pour tester votre parseur.

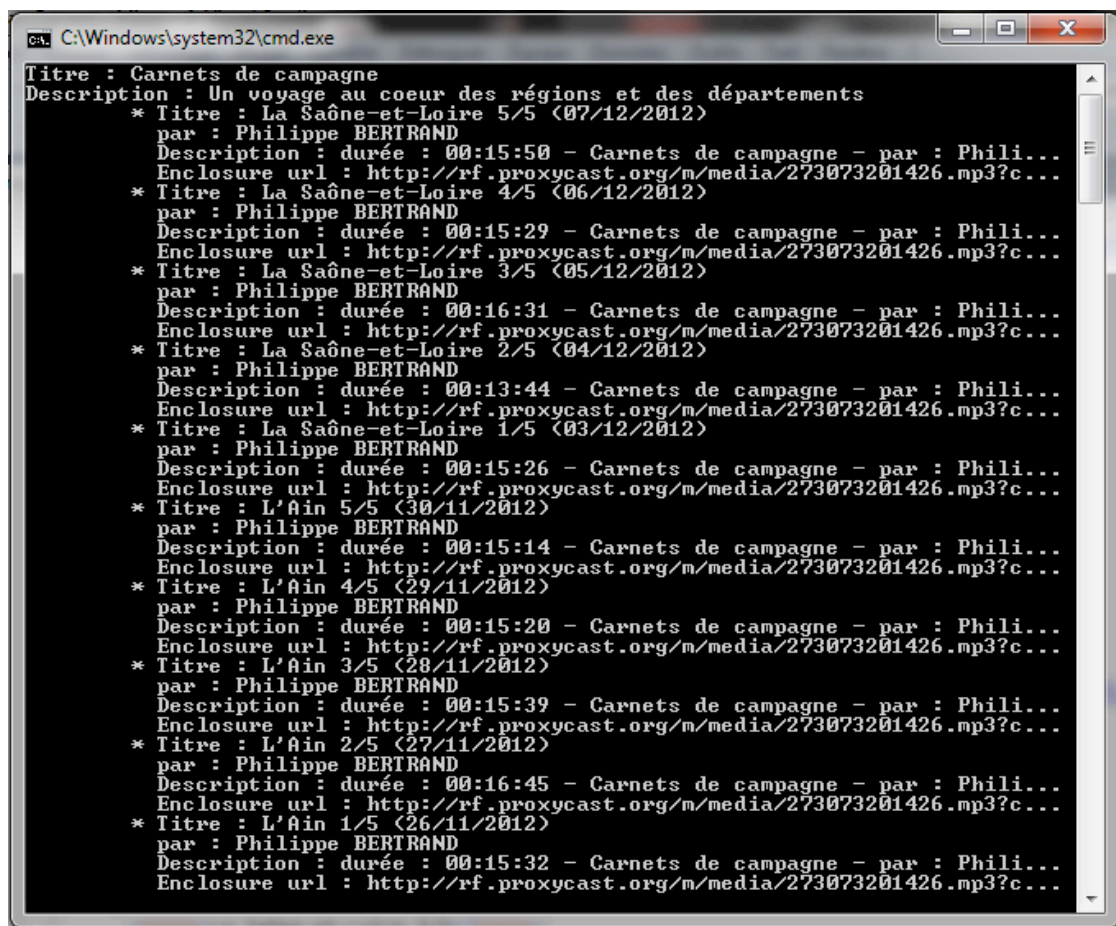
QUESTION 1 : LECTURE

Écrivez un parseur XML en C# permettant de lire les flux RSS fournis. Ce parseur devra lire :

- le titre du channel,
- la description du channel
- puis pour chaque item :
 - le titre de l'item,
 - la date de publication (pubDate),
 - l'auteur de l'item (celui de la balise <itunes:author>)
 - la description de l'item,
 - l'url (enclosure url)

Vous vérifierez le bon fonctionnement de votre parseur en affichant à l'écran le contenu du flux rss, à l'aide par exemple, des méthodes ToString de Channel et Item.

Pour le fichier rss_10205.xml, vous devriez obtenir le résultat suivant :



```
C:\Windows\system32\cmd.exe
Titre : Carnets de campagne
Description : Un voyage au coeur des régions et des départements
* Titre : La Saône-et-Loire 5/5 <07/12/2012>
  par : Philippe BERTRAND
  Description : durée : 00:15:50 - Carnets de campagne - par : Phili...
  Enclosure url : http://rf.proxycast.org/m/media/273073201426.mp3?c...
* Titre : La Saône-et-Loire 4/5 <06/12/2012>
  par : Philippe BERTRAND
  Description : durée : 00:15:29 - Carnets de campagne - par : Phili...
  Enclosure url : http://rf.proxycast.org/m/media/273073201426.mp3?c...
* Titre : La Saône-et-Loire 3/5 <05/12/2012>
  par : Philippe BERTRAND
  Description : durée : 00:16:31 - Carnets de campagne - par : Phili...
  Enclosure url : http://rf.proxycast.org/m/media/273073201426.mp3?c...
* Titre : La Saône-et-Loire 2/5 <04/12/2012>
  par : Philippe BERTRAND
  Description : durée : 00:13:44 - Carnets de campagne - par : Phili...
  Enclosure url : http://rf.proxycast.org/m/media/273073201426.mp3?c...
* Titre : La Saône-et-Loire 1/5 <03/12/2012>
  par : Philippe BERTRAND
  Description : durée : 00:15:26 - Carnets de campagne - par : Phili...
  Enclosure url : http://rf.proxycast.org/m/media/273073201426.mp3?c...
* Titre : L'Ain 5/5 <30/11/2012>
  par : Philippe BERTRAND
  Description : durée : 00:15:14 - Carnets de campagne - par : Phili...
  Enclosure url : http://rf.proxycast.org/m/media/273073201426.mp3?c...
* Titre : L'Ain 4/5 <29/11/2012>
  par : Philippe BERTRAND
  Description : durée : 00:15:20 - Carnets de campagne - par : Phili...
  Enclosure url : http://rf.proxycast.org/m/media/273073201426.mp3?c...
* Titre : L'Ain 3/5 <28/11/2012>
  par : Philippe BERTRAND
  Description : durée : 00:15:39 - Carnets de campagne - par : Phili...
  Enclosure url : http://rf.proxycast.org/m/media/273073201426.mp3?c...
* Titre : L'Ain 2/5 <27/11/2012>
  par : Philippe BERTRAND
  Description : durée : 00:16:45 - Carnets de campagne - par : Phili...
  Enclosure url : http://rf.proxycast.org/m/media/273073201426.mp3?c...
* Titre : L'Ain 1/5 <26/11/2012>
  par : Philippe BERTRAND
  Description : durée : 00:15:32 - Carnets de campagne - par : Phili...
  Enclosure url : http://rf.proxycast.org/m/media/273073201426.mp3?c...
```

Contraintes :

- utilisez des classes de stockage (celles qui vous sont fournies ou d'autres).

Aide :

Bien qu'il n'y ait pas de schéma, les flux rss sont toujours constitués de la même manière ou presque. Vous pourrez donc considérer dans votre parseur, pour simplifier, que les éléments et attributs recherchés existent toujours dans le fichier, mais peuvent apparaître dans n'importe quel ordre à l'intérieur d'un item ou d'un channel.

QUESTION 2 : ÉCRITURE

Modifiez le parseur précédent ou écrivez un nouveau parseur XML en C# permettant d'ajouter un nouvel item à un des flux rss précédents. Ce parseur permettra de rajouter un item simple contenant :

- un titre : «*MonTitre*»
- une date de publication : *maintenant*
- un auteur (<itunes:author> encore) : «*Moi*»
- une description : «*Ma description*»
- un url (enclosure url) : «*http://...*»

Votre parseur enregistrera le résultat dans un nouveau fichier XML («test.xml» par exemple) qui contiendra donc les anciens items et le nouveau.

Vérifiez avec le résultat de la question 1 que vous pouvez lire ce fichier et votre nouvel élément.

Note : il n'est pas nécessaire d'ajouter les autres éléments du flux qui ne sont pas lus.