

TP 2 : API SAX DOM et JAXP

Objectif

Ce TP a pour objectif de vous faire manipuler les APIs JAXP, SAX et DOM afin de charger un document XML et d'effectuer des traitements sur celui-ci.

Il se compose de 5 exercices

- Charger un Document XML avec DOM
- Parcourir un document XML
- Modifier un document XML
- Transformer un modèle de document avec une feuille de style et sauvegarder le résultat
- Lire un document XML avec SAX

Quelques rappels :

Vous serez amenés à manipuler plusieurs interfaces de développement XML :

- Interface DOM : Modèle orienté Objet pour des documents structurés
- Interface SAX : Modèle événementiel
- JAXP : API unifiée pour la création de parser DOM et SAX

Les outils que vous utiliserez sont :

- La JDK
- La librairie JAXP
- La librairie DOM (org.w3c.dom.*)
- La librairie SAX (org.w3c.sax.*)
- La librairie Xerces (xerces.jar disponible sur neptune /home/depot/3A/XML/TP2)

1) Charger un document DOM

Objectif :

Lecture d'un fichier XML et création du modèle de document correspondant

Outils :

- Librairie DOM

– Partie de JAXP

Travail :

Ecrivez une classe Java qui charge le document XML corrigé du TP précédent (disponible sur /home/depot/3A/XML/TP2) et qui instancie un document DOM.

Pour cela, utilisez JAXP pour instancier un parser XML et DOM pour obtenir le modèle de document correspondant.

Voici l'ordre des opérations à mener :

1. Définir un document DOM
2. Gérer les exceptions
3. Instancier un factory
4. Configurer le factory (éventuellement)
5. Obtenir un parser
6. Lancer le parser

2)Parcourir un Document DOM

Objectif :

Manipuler le document DOM obtenu

Travail :

Ajoutez une méthode permettant d'afficher l'ensemble des acteurs à l'écran. Il est possible d'effectuer ceci :

- a) en accédant directement aux éléments acteurs
ou
- b) en effectuant un parcours récursif des nodes de l'objet DOM (n'est pas forcément nécessaire dans votre cas puisque vous savez localiser précisément les acteurs grâce à la DTD)
ou
- c) en utilisant la methode getElementByTagName de document

Ajoutez une méthode permettant d'afficher l'ensemble des dates de retour.

3)Modifier un document DOM

Objectif :

Modifier un modèle de document en utilisant l'interface DOM.

Travail :

Ajoutez une méthode permettant de modifier le modèle de document lorsqu'un DVD est loué.

4)Transformer et écrire un Document XML à partir d'un modèle DOM

Objectif :

Sauver un modèle DOM et utiliser une feuille de style

Travail :

Sauvegardez le nouveau modèle de document dans un fichier XML et vérifiez que la méthode précédente fournit le résultat escompté (évitons la présence de deux emprunts simultanés sur un DVD par exemple).

Appliquez une des transformations écrites au TP précédent au modèle de document et sauvegardez le résultat dans un nouveau fichier.

5)Lire un Document avec SAX

Objectif :

Utiliser un parser SAX

Travail :

Créer une classe JAVA pour instancier un parser SAX.

Traitez le document XML pour afficher les noms des films contenus dans le document XML .

Outils :

Utilisez le package SAX

Utilisez le parser xerces (org.apache.xerces.parsers.SAXParser)

- soit directement dans le source
- soit en modifiant la variable système org.xml.sax.parser au lancement de l'application (option -D de java)

Pensez à mettre à jour votre classpath !!