

Feuille de travaux pratiques n° 1

XSLT, SAX & DOM

Exercice 1 Assemblée Nationale

Le thème du TP est issu de l'Open Data de l'assemblée nationale (<http://data.assemblee-nationale.fr/>). On s'intéresse plus particulièrement au jeu de données "DÉPUTÉS-SÉNATEURS-MINISTRES DE LA LÉGISLATURE"¹. Ce jeu comprend les données relatives à l'état civil des ministres, députés et sénateurs élus lors de la XVème législature en juin 2017, à leur appartenance aux organes parlementaires, aux assemblées parlementaires internationales et à des organismes extra-parlementaires où l'Assemblée nationale est représentée ainsi que leur rattachement à un parti politique. On s'intéresse aussi aux différents scrutins dans le jeu "VOTES"².

Ces données sont disponibles en format XML et JSON. Pour simplifier le traitement, toutes les données ont été regroupées dans un seul et même document disponible sur Madoc intitulé "assemblee1920.xml".

On cherche à écrire un programme permettant d'obtenir les députés qui ont voté "pour" pour un scrutin dont le titre contient "l'information". Les personnes sont dans l'ordre alphabétique des noms et des prénoms. On donnera, pour chaque vote, le groupe parlementaire d'appartenance, la date du scrutin, le résultat du scrutin et la présence (ou non) du député. Le document à générer doit respecter la DTD «info.dtd» suivante :

```
1 <?xml encoding="UTF-8"?>
2
3 <!ELEMENT information (act)*>
4
5 <!ELEMENT act (sc)+>
6 <!ATTLIST act nom CDATA #REQUIRED>
7
8 <!ELEMENT sc EMPTY>
9 <!ATTLIST sc
10   nom CDATA #REQUIRED
11   sort (adopté|rejeté) #REQUIRED date CDATA #REQUIRED
12   mandat CDATA #REQUIRED grp CDATA #REQUIRED
13   présent (Oui|Nom) #REQUIRED>
```

Sur le document "assemblee1920.xml", cela donnera le document "info.xml" présent sur Madoc.

Question 1

Donner le programme permettant d'obtenir le résultat précédent en utilisant XSLT, sans utiliser «xsl:for-each» ou des règles nommées.

Question 2

Donner les programmes PHP permettant d'obtenir le résultat précédent en utilisant les outils suivants : SAX³, DOM sans utiliser XPath (avec au moins une version sans utiliser "getElementsByTagName()"), DOM en utilisant XPath (sans utiliser «//»). Pour chaque version, faites en sorte que le traitement soit optimal en temps de traitement.

Question 3

Discuter des différentes versions proposées (XSLT, DOM, DOM/XPath, SAX) en termes de performance, d'adéquation au traitement, de clarté du code, de maintenance, etc.

Question 4 Facultative

Reprendre cette recherche en utilisant :

1. Java et JDOM. Pour cela, vous pourrez vous appuyer sur le site Web de JDOM. Deux versions sont attendues : avec et sans filtres. Là encore, le résultat sera construit en JDOM.

1. <http://data.assemblee-nationale.fr/acteurs/deputes-senateurs-ministres>
2. <http://data.assemblee-nationale.fr/travaux-parlementaires/votes>
3. Vous pouvez utiliser la bibliothèque Sax4PHP : <http://sourceforge.net/projects/sax4php>

2. les objets PHP : XMLReader pour le traitement et XMLWriter pour la génération. Comparez avec les solutions précédentes.
3. l'API SimpleXML de PHP.

Travail à rendre :

Vous devrez rendre votre travail à l'enseignant sous forme électronique d'une part le rapport (un petit document PDF de 5 à 7 pages présentant votre travail, analyses, algorithmes, conclusions, etc., les subtilités de vos traitements et vos difficultés) et d'autre part une archive de nom "noms-binome.tar.gz" où vous donnerez :

- votre code (bien présenté et commenté) ;
- un fichier "readme" et un fichier shell permettant à l'enseignant d'exécuter votre code sur sa machine.

Deux zones de dépôt sur Madoc seront disponible à cet effet. Une séance de tests pourra être organisée.

Quelques références :

- Standard XPath : <http://xmlfr.org/w3c/TR/xpath>
- Standard DOM : <http://www.w3.org/DOM/>
- Documentation XMLLINT : <http://xmlsoft.org/xmllint.html> ;
- Documentation XSLTProc : <http://xmlsoft.org/XSLT/xsltproc.html>
- Documentation PHP DOM : <http://www.php.net/manual/fr/ref.dom.php>
- Documentation PHP «SAX» : <http://www.php.net/manual/fr/ref.xml.php>
- Documentation PHP SimpleXML : <http://php.net/manual/fr/book.simplexml.php>
- Documentation XMLReader/XMLWriter : <http://fr2.php.net/manual/fr/book.xmlreader.php>
& <http://fr2.php.net/manual/fr/book.xmlwriter.php>
- Site JDOM : <http://www.jdom.org/downloads/docs.html>

RAPPEL :

- Il est aussi possible d'exécuter du PHP dans une fenêtre «terminal» en faisant : `<php mon_exo.php>`.
- Il est possible d'exécuter le code XSLT en ligne de commande en utilisant Saxon⁴, par exemple : `<java -jar saxon9he.jar -xsl :mon_exo.xsl -s :mondial.xml>`. L'ajout du paramètre `"-TP :profile.html"` permettra d'obtenir un profilage de l'exécution en HTML. XSLT peut aussi être exécuté par PHP.
- Éventuellement, pour évaluer le temps de traitement, il faut faire précéder la ligne par `<time>`, par exemple : `<time php mon_exo.php>` ou `<time java MaClasse>`. Le temps de traitement effectif de votre programme est la ligne `<user>`.

4. Documentation : <http://www.saxonica.com/documentation/> . Téléchargement : <https://sourceforge.net/projects/saxon/files/>.