
WEB DES DONNÉES

TP N° 4 – XSLT

Romain LELONG – romain.lelong@gmail.com

Avant de commencer ...

Outre le présent sujet, les fichiers suivants doivent être récupérés avant de débiter le TP :

- `le-seigneur-des-anneaux-1.jpg`
- `le-seigneur-des-anneaux-2.jpg`
- `le-seigneur-des-anneaux-3.jpg`
- `sda.xml`
- `star.png`
- `style.css`

Ce TP ne sera pas noté et aucun compte-rendu n'est demandé. Vous pouvez le réaliser en binôme ou seul. Les divers éléments abordés dans celui-ci pourront néanmoins faire l'objet d'une micro-évaluation prochaine.

1 Firefox & XSLT, les prérequis ...

Pour des raisons de sécurité, les nouvelles versions de Firefox ne permettent plus d'appliquer les feuilles de style XSLT aux fichiers stockés localement sur la machine. En d'autres termes, lorsque Firefox accède à un fichier XML par l'intermédiaire d'une adresse du type `file://...`, la feuille XSLT qui est référencée dans ce fichier ne peut plus être appliquée.

Afin de pallier à ce problème, il est nécessaire d'accéder au fichier XML par l'intermédiaire du protocole HTTP c'est à dire de posséder une URL du type `http://...` pour y accéder.

Vous pouvez utiliser le serveur HTTP de votre choix. Néanmoins, il est proposé, dans ce TP soit :

- d'utiliser le conteneur web de servlets et JSP Tomcat ;
- de démarrer un serveur Web local à l'aide de Python3.

Il est également possible de désactiver certains paramètres de sécurité dans la configuration de Firefox. Mais cette méthode étant non sécurisée, elle ne sera pas abordée.

1.1 Première méthode : Utiliser le serveur Tomcat

Tomcat nécessite que Java soit installé et configuré sur votre machine. Si vous travaillez sur les machines de l'université, c'est déjà le cas mais si vous travaillez sur votre machine personnelle, assurez vous que :

- La version 11 de Java (ou supérieure) est installée sur votre machine (`java -version` dans un terminal) ;
- Que la variable d'environnement `JAVA_HOME` est positionnée (`echo $JAVA_HOME` dans un terminal).

Voici la démarche à suivre :

1. Télécharger la dernière version de Tomcat 10 à partir du site de Tomcat (i.e. <https://tomcat.apache.org/>). L'archive doit porter un nom du type `apache-tomcat-10.x.xx.zip`
2. décompresser cette archive à l'endroit de votre choix.
Attention : Ne pas décompresser l'archive sur un dispositif de stockage externe (e.g. clé USB) ni sur votre ENT.
3. Créer un sous-répertoire TP4 dans le répertoire `./apache-tomcat-10.x.xx/webapps/`. Ce répertoire sera dédié à votre TP . C'est dans ce dernier que vous pourrez déposer vos fichiers XML et XSLT afin de les rendre accessibles via HTTP par l'intermédiaire d'une URL du type :

`http://localhost:8080/TP4/fichier`

où *fichier* doit être remplacé par le nom complet (i.e. avec son extension) du fichier.

4. Le serveur Tomcat possède un interface graphique appelé « *manager* » permettant de gérer les diverses applications Web présentes dans le serveur. Ce manager est sécurisé par authentification. Pour pouvoir y accéder il est nécessaire de paramétrer un utilisateur, son mot de passe et son rôle dans le fichier `.../apache-tomcat-10.x.xx/conf/tomcat-users.xml`. Pour cela il faut y ajouter, dans la partie gérant les rôles et utilisateur le code suivant :

```
<role rolename="tomcat"/>
<user username="tomcat" password="tomcat" roles="tomcat,manager-gui"/>
```

5. Le serveur Tomcat peut être démarré à l'aide du script `startup.sh` (resp. `startup.bat`) pour linux (resp. Windows) situé dans le répertoire `.../apache-tomcat-10.x.xx/bin/`. De même, il le serveur peut être arrêté à l'aide du script `shutdown.sh` (resp. `shutdown.bat`) pour Linux (resp. Windows).
Remarque : Si vous travaillez sur votre machine Linux, et que vous ne disposez pas des droits nécessaires pour exécuter le script `startup.sh` ou `shutdown.sh`, vous pouvez utiliser les commandes suivantes pour pallier à ce problème :

```
sudo chown -R votre user ./apache-tomcat-10.x.xx
sudo chmod -R gu+x ./apache-tomcat-10.x.xx
```

1.2 Deuxième méthode : démarrer un serveur Web local à l'aide Python3

Il est très simple avec Python3 de démarrer un serveur Web local afin de rendre le contenu d'un dossier accessible via HTTP. Pour cela :

1. Se rendre dans le répertoire contenant les fichiers XML et XSLT que l'on souhaite rendre accessible via HTTP.
2. Utiliser la commande suivante pour lancer le serveur :

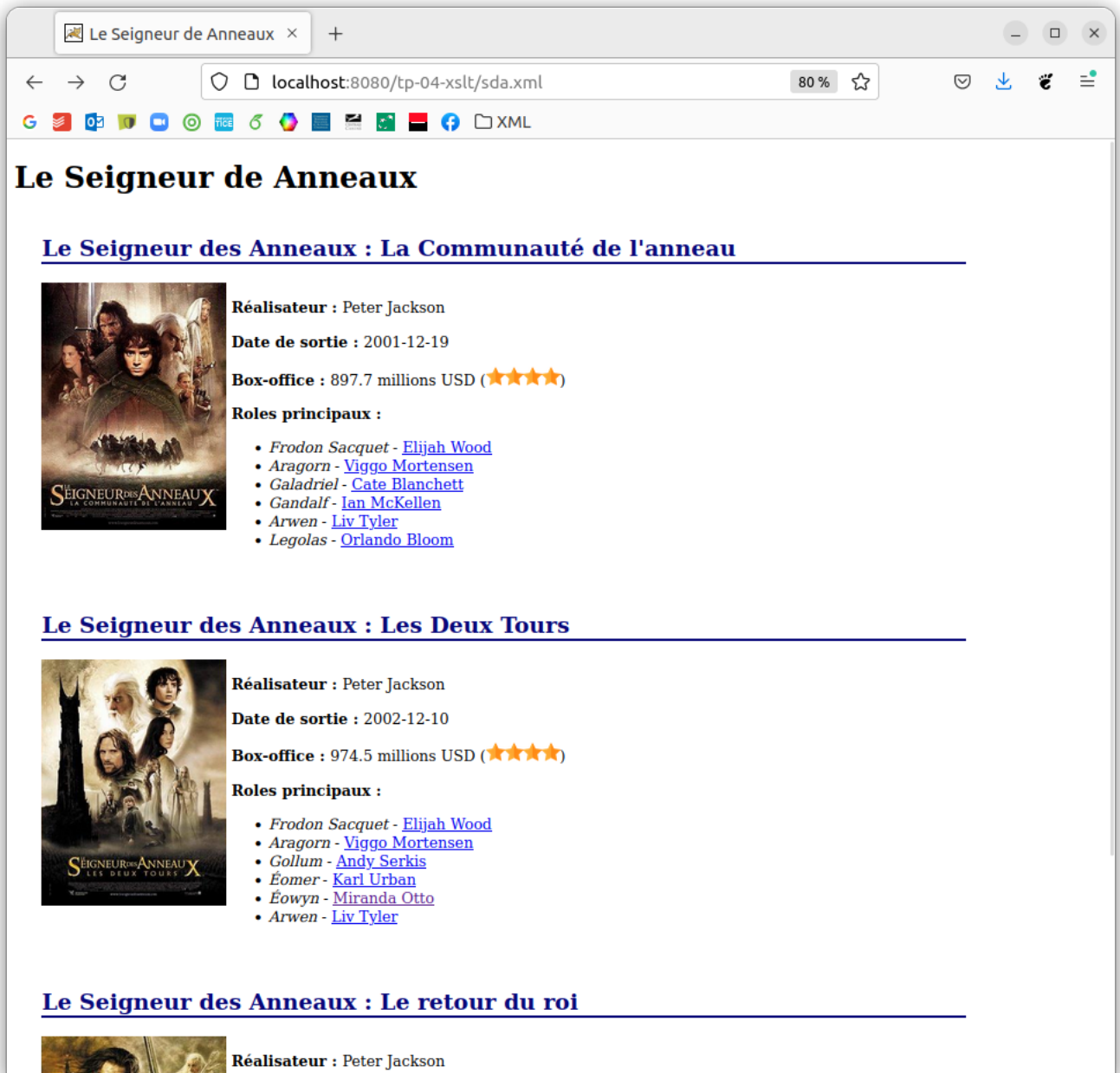
Linux : `python3 -m http.server`

Windows : `python -m http.server` ou bien `py -3 -m http.server`

Par défaut, le contenu du répertoire sera rendu disponible via un serveur web local, sur le port 8000. Vous pourrez alors accéder au fichier contenu dans le dossier par l'intermédiaire d'une URL du type : `http://localhost:8000/fichier`

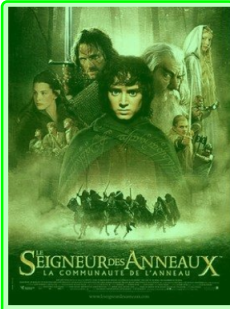
2 Objectif du TP

Le document XML `sda.xml` fourni avec ce TP contient divers informations (au format XML) sur les trois films de la trilogie *Le Seigneur des Anneaux*. Dans ce TP il vous est demandé de réaliser une feuille de style `sda.xslt` permettant de transformer ce fichier afin d'en donner une présentation au format HTML. L'objectif final est d'aboutir à une présentation des données identique à la capture d'écran suivante :



L'objectif de ce TP étant de concevoir une feuille de style XSLT et non de maîtriser les subtilités du CSS et du HTML, la trame de la page HTML resultat vous est donnée ci-après. Par ailleurs, la feuille de style CSS `style.css` s'appliquant à cette trame vous est également fournie. Si vous respectez la trame HTML, vous n'aurez pas à vous préoccuper du CSS.

Le Seigneur des Anneaux : La Communauté de l'anneau



Réalisateur : Peter Jackson

Date de sortie : 2001-12-19

Box-office : 897.7 millions USD (★★★★★)

Roles principaux :

- Frodon Sacquet - [Elijah Wood](#)
- Aragorn - [Viggo Mortensen](#)
- Galadriel - [Cate Blanchett](#)
- Gandalf - [Ian McKellen](#)
- Arwen - [Liv Tyler](#)
- Legolas - [Orlando Bloom](#)

```
<html>
<head>
  <link href="style.css" rel="stylesheet" type="text/css">
  <title>...</title>
</head>
<body>
  <h1>...</h1>
  <div class="film-div">
    <div class="film-title-div">
      <h2>...</h2>
    </div>
    <div class="film-content-div">
      <div class="film-left-div">
        
      </div>
      <div class="film-right-div">
        <p>
          <b>... : </b> ...
        </p>
        <p>
          <b>... : </b>...
        </p>
        <p>
          <b>... : </b>...
        </p>
        <b>... :</b>
        <ul>
          <li><i>...</i> - <a href="...">...</a></li>
          <li><i>...</i> - <a href="...">...</a></li>
          <li><i>...</i> - <a href="...">...</a></li>
          <li><i>...</i> - <a href="...">...</a></li>
          <li><i>...</i> - <a href="...">...</a></li>
        </ul>
      </div>
    </div>
  </div>
</div>
</body>
</html>
```

3 Élément requis

Les points suivants doivent impérativement être respectés :

1. Aucun texte apparaissant dans la page HTML resultat ne doit être « écrit en dur » dans la feuille de style XSLT `sda.xslt` excepté pour le mot clé « millions » et les signes de ponctuation (e.g. « : », « - », « (» etc.). Ce texte doit en effet être systématique récupéré dans le fichier soucre `sda.xml` à l'aide d'éléments XSL.
2. Les gains au Box-Office doivent, comme la copie d'écran le montre, être affichés en millions de dollard (USD) et non en dollard (USD). Une conversion doit donc être réalisée dans la feuille de style XSLT.
3. Les petites étoiles permettent d'illustrer graphiquement les gains au Box-Office. La règle suivante doit être mise en application dans la feuille de style XSLT :
Si *gain* > 1000 : 5 étoiles ;
Si *gain* > 500 : 4 étoiles ;
Si *gain* > 100 : 3 étoiles ;
Si *gain* > 50 : 2 étoiles ;
Sinon : 1 étoiles.
4. Le nom de acteurs doivent être de liens hypertexte vers la page Wikipedia de l'acteur en question.
5. La feuille de style XSLT ne doit contenir aucune boucle `xsl:for-each`. Il est donc nécessaire d'exploiter les élément `xsl:apply-templates` à la place.
6. Plus généralement, une très grande importance sera accordée à la modularité de votre feuille de style. En d'autres termes, il est fortement conseillé de répartir, dès que cela est possible et cohérent, les divers traitements et transformations XSLT dans différents templates plutôt que de tous les agréger et/ou les répéter.