Sérialisation

Laboratoire n°3 Christophe Peretti Samuel Darcey

Table des matières

[1 Introduction 2](#_Toc449698341)

[2.1 XML 2](#_Toc449698342)

[2.2 JSON 4](#_Toc449698343)

[3 Conclusion 5](#_Toc449698344)

# 1 Introduction

Le laboratoire SER – Plex consistant à simuler les activités d’un complexe cinématographique, est constitué de 4 laboratoires constituant les différentes parties du projet. Le présent document porte sur le troisième laboratoire : Construction de fichier html avec xsl, et utilisation de css.

Dans ce laboratoire, il s’agissait concevoir un fichier xsl, qui prendrait le fichier xml généré dans le deuxième laboratoire, et qui en générerais un ou plusieurs fichiers html, afin d’avoir des pages web représentant le cinéma.

# 2 Modification du fichier xml

Dans le précédent laboratoire, nous avions décidé d’épurer le xml afin de ne garder que les quelques champs les plus utiles. Au final, il nous manquait plusieurs informations essentielles qui devraient figurer sur un site web de cinéma, comme les notes et les critiques. Dans ce laboratoire, nous avons commencé par modifier notre fichier java de génération du xml, afin que celui-ci comporte toutes les balises demandées.

# 3. Implémentation

## 3.1 Génération des pages

Pour ce laboratoire, nous avons décidé de générer une seule page principale ayant les projections de film triées par date, titre et note moyenne. Pour les descriptions de film, nous générons une page par film.

Sur la capture suivante, ce trouve en premier la génération du fichier index.html, qui appelle les trois templates permettant de sélectionner les projections par date, titre ou note (ces templates se trouvent plus bas dans le code). Nous avons décidé de mettre ces trois formes de tri sur une seule page, afin de ne pas multiplier le nombre de page html générée. Une autre possibilité aurait été, grâce à du javascript, de générer du contenu dynamique en fonction du choix de tri de l’utilisateur.



Dans la deuxième partie, nous pouvons voir la génération des pages html des films, ayant le titre de celui-ci dans le nom du fichier. Ces fichiers comprennent l’ensemble des informations d’un film.

## 3.2 Templates de tri

Voici les trois templates de tri des projections. Nous pouvons voir qu’ils sont presque identiques, les seules différences sont le titre et la méthode de tri. Dans les trois cas nous appliquons le template qui s’applique sur film en mode tableau.



## 3.3 Affichage du tableau

Dans l’affichage des tableaux de projections, nous allons simplement chercher dans le xml les dates de projection, le numéro de la salle du cinéma, le titre du film correspondant ainsi que sa durée et la note moyenne des critiques.

## 

L’entête des tableaux était appelé avant le parcours des projections, dans les templates précédents. Nous pouvons voir que le titre du film est encadré d’une balise <a></a>, qui permet de créer un lien hypertexte. La référence de ce lien est le fichier html du film en question. Il est donc possible d’accéder à ce fichier en cliquant directement sur le film présent dans une projection.

## 3.4 Fichier de film

## 2.2 JSON

Par rapport au premier laboratoire, la génération du fichier Json a aussi légèrement été modifiée. Nous n’avons plus de catégorie « film » mais directement le titre de celui-ci dans la projection. Idem avec les acteurs. Pour la génération du fichier Json, nous utilisons un GsonBuilder dans lequel nous ajoutons des JsonObject ou des JsonArray. Nous avons un array pour els projections et un pour les roles. Bien sûr, dans notre Json nous affichons uniquement les deux premiers rôles du film. Ces rôles peuvent être dans le désordre (place 2 avant la 1) car ces données ne sont pas triées dans les globalData.

# 3 Conclusion

Malgré quelques différences, nous avons tout de même réussi à générer des fichiers xml et json relativement similaires à ceux du premier laboratoire. L’utilisation d’une base de données Mysql nous a aussi mis en situation réelle dans l’utilisation de ces fichiers xml et json.