SER – Laboratoire 1

Table des matières

[Introduction 1](#_Toc447803982)

[XML 1](#_Toc447803983)

[Exemple 1](#_Toc447803984)

[DTD 4](#_Toc447803985)

[JSON 4](#_Toc447803986)

[Conclusion 4](#_Toc447803987)

# Introduction

Le but de ce laboratoire est de mettre en place les structures XML et JSON qui seront nécessaires pour la réalisation des prochains laboratoires.

# XML

Le fichier XML se présente comme un document composé de 5 sections :

* les films,
* les acteurs,
* les genres,
* les langues,
* les salles

Dans le cas des films, nous avons également 4 sous-sections :

* les rôles,
* les projections,
* les mots-clés,
* les critiques

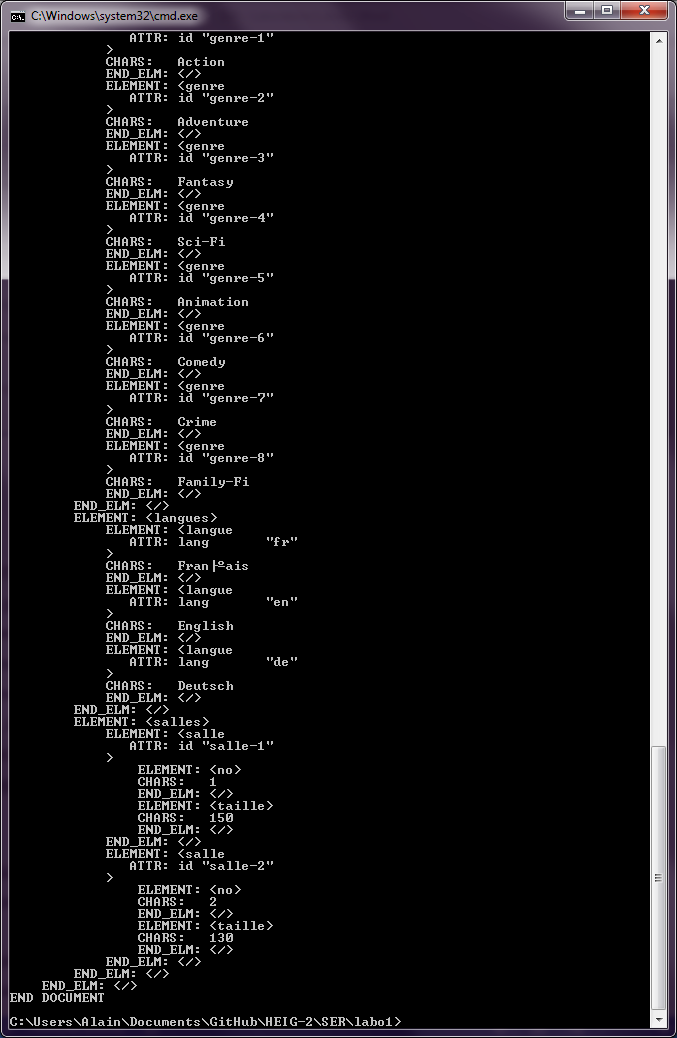
Les informations que nous avons choisi de représenter comme une sous-section de *film* sont celles qui sont fortement liées au *film* et qui n’auraient pas de sens dans le contexte d’un autre *film*. Puisqu’il n’est pas raisonnable de les partager entre plusieurs *films*, ils sont donc des sous-éléments de *film*, matérialisant leur relation avec le *film* parent.

À l’inverse, un même *acteur* peut jouer dans un grand nombre de *films*. Il est ainsi bien plus pratique de le définir qu’une seule fois et d’effectuer un lien entre le *film* et l’*acteur*. De même pour genres, langues et salles qui sont des ensembles de valeurs partagées par tous les films.

### Exemple

Voir annexe XML.xml

### Validation



## DTD

Le document DTD impose les contraintes structurelles suivantes :

* Un film peut ne pas avoir d’*acteur*. Nous avons donc défini les *rôles* comme optionnels.
* De même pour *projections*, un *film* peut ne jamais être projeté.
* De même pour *critiques*, un *film* peut n’avoir aucune critique
* *Genres*, *langues* et *salles* sont des données externes aux *films*, ils sont donc référencés par leur ID. De même pour les *acteurs*.
* Les *images* des films sont représentées sous forme de liens vers des fichiers externes.

### Grammaire

Voir annexe XML.dtd

# JSON

Le document JSON est beaucoup plus simple et ne contient qu’un sous-ensemble des informations contenues dans le XML.

Nous avons choisi d’utiliser un élément racine de type liste représentant la liste des projections. Chaque projection est une structure contenant les champs *nom*, *date* et *roles*, ce dernier étant une liste de chaines de caractère correspondant au nom des acteurs.

À noter que si nous souhaitions rajouter d’autres informations que les projections, il serait nécessaire de modifier le type de l’élément racine en structure et de placer la liste des projections dans un champs *projections* de cette structure.

### Exemple

Voir annexe JSON.json

# Conclusion

Nous constatons rapidement que le document XML est défini de façon bien plus formelle à travers sa DTD que le document JSON pour lequel il n’existe pas de langage de spécification officiel. De cette façon, un programme consommant un document XML sera probablement plus fiable puisque la forme des données est strictement définie et qu’il est possible de valider facilement la structure du document avec des outils existant utilisant la DTD.

Nous pouvons cependant mentionner le projet JSON-Schema qui défini une méthode pour décrire la structure des documents JSON de façon similaire à ce qu’une DTD fait pour un document XML. En utilisant JSON-Schema, une application JSON peut bénéficier des mêmes avantages qu’une application XML utilisant les DTDs.