SER – Laboratoire 1

HEIG-VD

Adriano Ruberto & Matthieu Villard

2016

Table des matières

[1. Introduction 2](#_Toc447742726)

[2. Structure du document XML 2](#_Toc447742727)

[3. Grammaire DTD 4](#_Toc447742728)

[4. Exemple de document XML 5](#_Toc447742729)

[5. Structure du document JSON 7](#_Toc447742730)

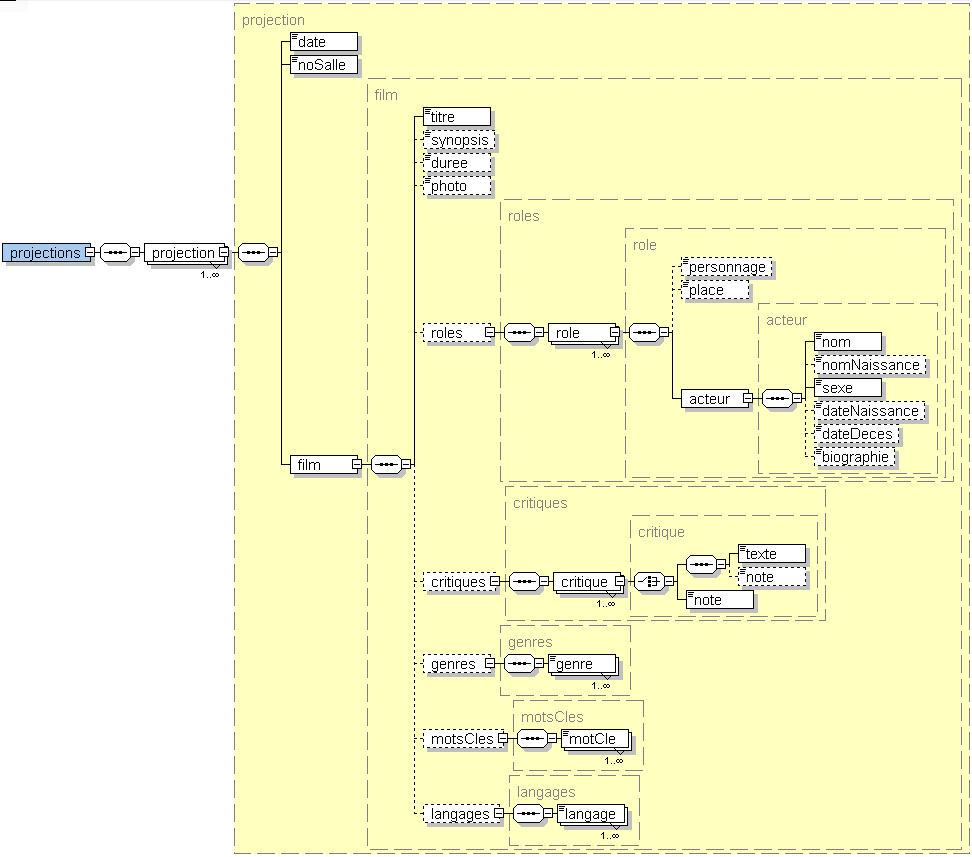
[6. Exemple de document JSON 7](#_Toc447742731)

[7. Conclusion 8](#_Toc447742732)

# ****Introduction****

Ce laboratoire a pour but de nous familiariser avec les bases de la sérialisation des données et plus particulièrement avec la normalisation des données. Nous avons donc tenté de normaliser les données contenues dans un document xml au moyen d’une grammaire DTD tel que demandé. A noter que, ayant également quelques notions au sujet de la grammaire XSD, nous avons également élaboré une grammaire XSD afin de comparer l’efficacité de ces deux grammaires.

# ****Structure du document XML****



Le document xml doit comprendre le détail des différentes projections afin d’être échangé avec des revues spécialisées, d’autres complexes cinématographiques ou encore des clients WEB (en effectuant une transformation en HTML).

Pour réaliser cet échange d’informations, il a donc été décidé d’utiliser un seul fichier xml contenant la liste des différentes projections. Toutes les informations sont regroupées par projection et par conséquent les projections représentent l’élément racine du document xml. De plus, il a été choisi de ne partager que le strict minimum d’informations. Ainsi, aucun identifiant, propre à la base de données, n’est divulgué.

Comme illustré dans la figure ci-dessus, les relations du type « un à plusieurs » sont systématiquement encapsulées dans un élément parent. Ainsi, la liste des genres, contenus dans l’élément *genre,* sont encapsulées dans l’élément parent *genres*. Il est vrai que cette méthodologie n’est pas triviale. Cependant, elle permet d’obtenir, depuis l’élément parent, uniquement les éléments désirés sans se soucier pour un traitement ultérieur.

L’observation de la description de la base de données d’où sont tirées les informations permet de constater qu’aucune donnée, hormis les clés, n’est obligatoire. Pour l’implémentation de l’échange d’informations réalisé dans le cadre de ce laboratoire, nous avons décidé d’ajouter certaines contraintes pour garantir la présence d’un minimum d’informations. Ainsi, les contraintes sont listées ci-dessous :

* Une projection doit avoir au minimum une date, un numéro de salle et un film. Il n’y aurait aucun sens de fournir une projection avec l’une de ces informations manquante.
* Un film doit avoir au minimum un titre. Sans cette information, il serait difficile d’identifier clairement un film. Nous nous sommes interrogés sur la pertinence de la possibilité pour un film de n’avoir aucun rôle. Puis, il nous est apparu qu’un documentaire sans commentaires oraux, pouvait correspondre à ce cas de figure.
* Un rôle doit avoir au minimum un acteur. La « place du rôle » d’un acteur n’est pas forcément pertinente et est donc optionnelle. Le personnage joué par un acteur est également optionnel, car un acteur peut prêter sa voix à un documentaire.
* Un acteur doit avoir au minimum un nom et un sexe. Les autres informations peuvent ne pas être connues ou ne pas exister (date de décès).
* Une critique peut être composée d’une note, d’un texte ou d’une note et d’un texte. En effet, nous avons considéré qu’une critique peut être formulée sans attribuer de note. De même une note peut être attribuée sans formuler de critique.

Rappelons que nous avons choisi de centraliser les informations pour chaque projection. Ce faisant, nous avons introduit une certaine redondance d’informations puisque, par exemple, un film peut faire l’objet de plusieurs projections. Nous avons opté pour cette solution afin de simplifier l’échange d’information. De plus, ce choix se justifie par le fait que les destinataires n’utilisent les informations qui leur sont transmises que pour établir un horaire des projections. Et pour éliminer la redondance, il aurait fallu partitionner les informations, ce qui nous paraissait fastidieux.

# Grammaire DTD

Pour illustrer au mieux le schéma xml, nous avons utilisé XMLSpy développé par Altova. Pour pouvoir obtenir le schéma ci-dessus, nous avons élaboré un schéma xml au format xsd. Puis, nous avons créé la grammaire DTD, demandée pour ce laboratoire, en nous inspirant de la grammaire XSD. En testant la validation de notre fichier xml d’exemple, nous avons pu constater que la grammaire XSD permet d’interdire la présence de balise vide alors que la grammaire DTD en est incapable. De plus, nous avons pu être témoins de la richesse du format XSD par rapport au DTD, beaucoup plus basique.

La figure ci-dessous illustre la grammaire DTD contenue dans le fichier *Labo01.dtd*.



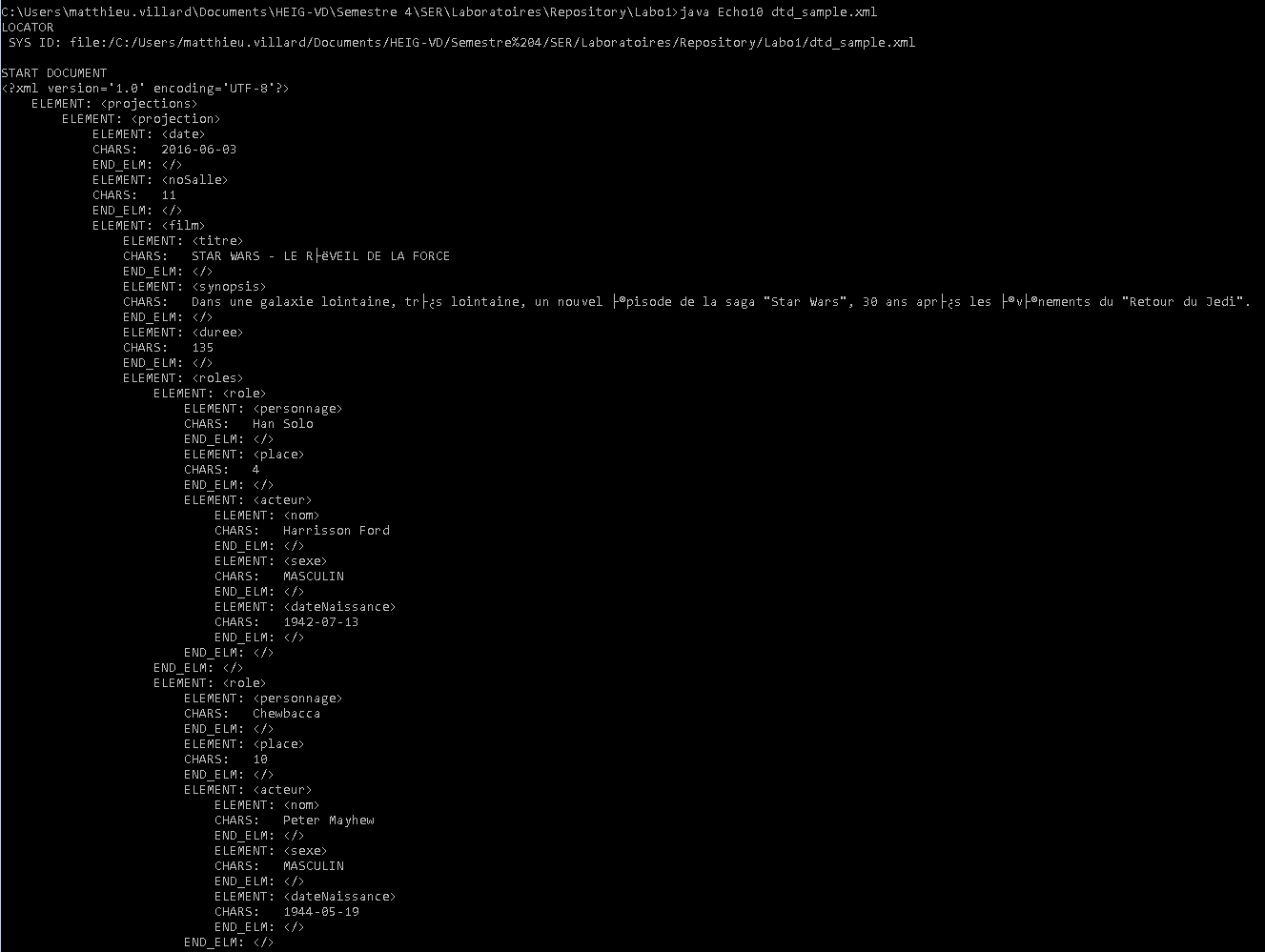
A titre de comparaison, la grammaire XSD, que nous avons réalisée en parallèle, occupe 126 lignes. Cette différence s’explique notamment par la possibilité d’utiliser des types dans une grammaire XSD et donc de réaliser une grammaire beaucoup plus précise.

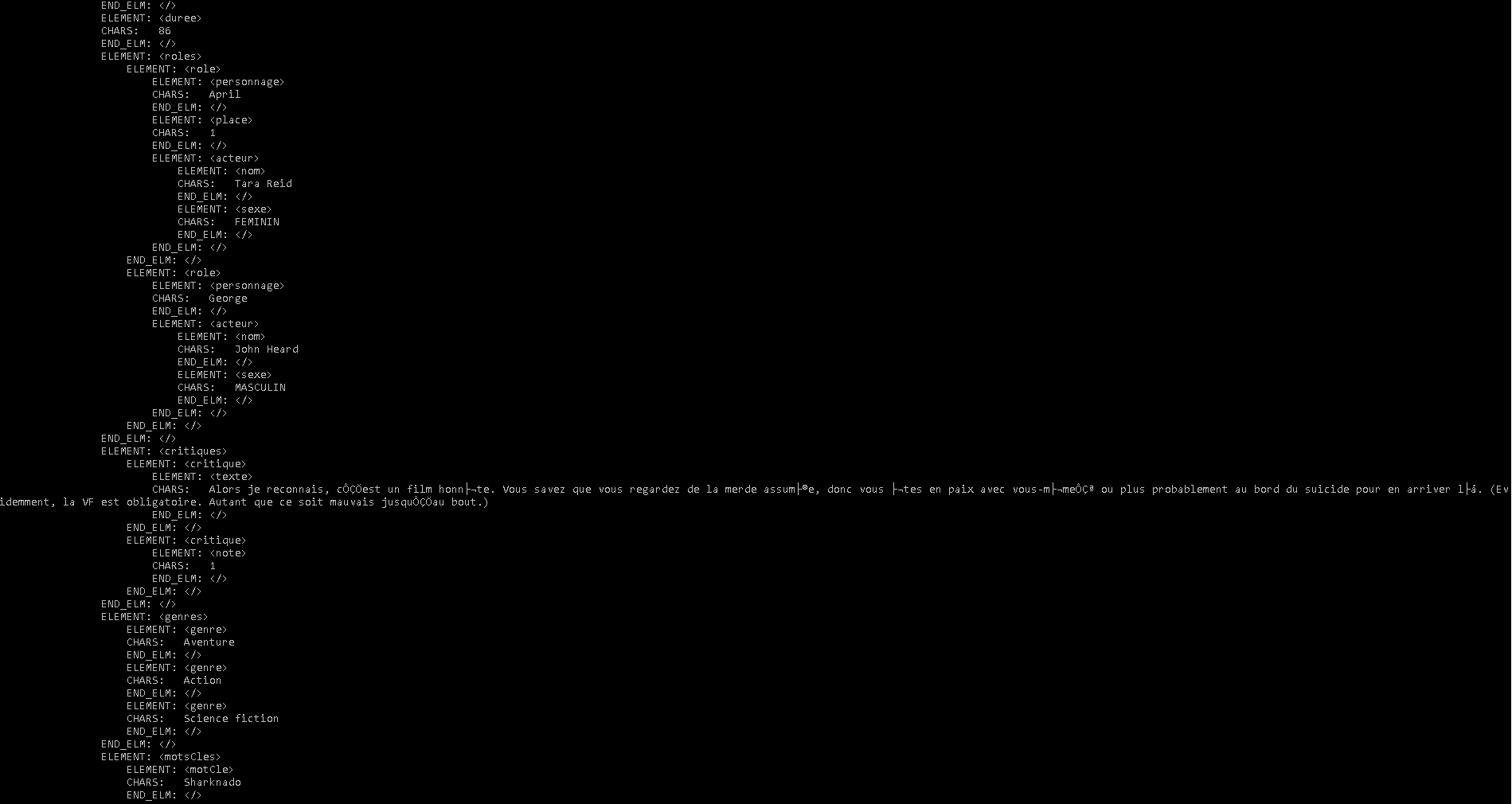
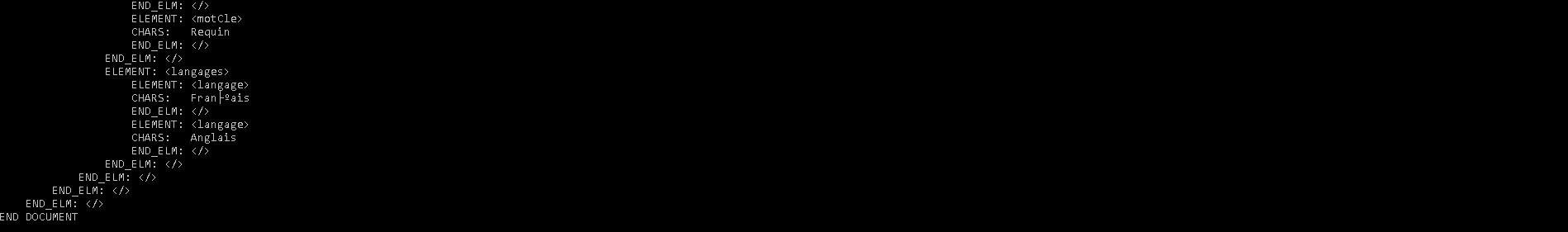
Si nous observons la grammaire DTD ci-dessus, nous pouvons constater que nous partons d’un élément racine de type complexe et que les éléments terminaux sont de type *#PCDATA*. Ainsi, il n’est pas possible de préciser clairement le type des données et de valider une date par exemple ou de préciser un types *ENUM* pour le sexe. C’est pourquoi, l’application qui utilisera le fichier xml que nous lui transmettrons devra, après validation du fichier, effectuer un traitement supplémentaire pour valider le format des données. Nous vous fournissons en annexe les fichiers contenant les grammaires DTD et XSD.

# Exemple de document XML

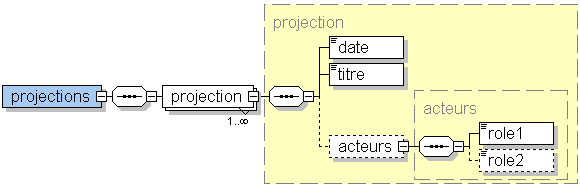
Nous vous fournissons en annexe deux documents xml à titre d’exemple, *dtd\_sample.xml* et *xsd\_sample.xml* pour illustrer un exemple de document xml correspondant respectivement à la grammaire DTD et la grammaire XSD.

Ci-dessous, vous trouverez le résultat de la validation du fichier dtd\_sample.xml au moyen du programme *echo10.java*. Malheureusement, le fichier d’exemple était trop volumineux pour que le résultat de la validation puisse être affiché en entier sur une seule page. En annexe, nous vous fournissons le programme *echo10*.



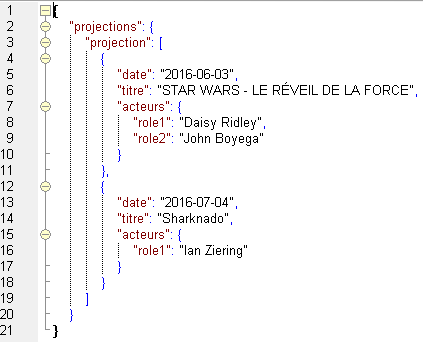
# Structure du document JSON



Le document json doit comprendre un résumé des différentes projections afin d’être échangé avec les médias.

Pour réaliser cet échange d’informations, il a donc été décidé d’utiliser un seul fichier json contenant la liste des différentes projections. Toutes les informations sont regroupées par projection et par conséquent les projections représentent l’élément racine du document json. Presque toutes les données sont obligatoires. Seuls les noms des acteurs sont optionnels. En effet, il est possible pour un film de n’avoir aucun acteur. De plus, un film peut n’avoir qu’un seul acteur et dans ce cas celui-ci occupe le 1er rôle.

# Exemple de document JSON



# Conclusion

Ce laboratoire nous a permis d’avoir un bref aperçu des méthodes de normalisation des données dans les documents xml.

Nous avons ainsi pu constater que l’utilisation d’une grammaire DTD permet de garantir la présence de certaines informations dans le document. Cependant, celle-ci ne garantit en aucun cas le format de ces données. De plus, même si nous obligeons la présence d’une information, sous la forme d’une balise, celle-ci peut être renseignée mais laissée vide. Ces problèmes sont propres à la grammaire DTD. En effet, la grammaire XSD permet de remédier à ces problèmes en introduisant la notion de types et en permettant de spécifier le format des données.

Nous relevons toutefois quelques difficultés rencontrées lors de ce laboratoire. En effet, l’utilisation de document xml pour l’échange ou le stockage de données ne nous étant pas encore totalement familier, la plus grosse difficulté était de concevoir la structure du document xml. Nous avons notamment pu nous interroger sur l’importance d’éviter la redondance de données. De plus, notre travail devant être utilisé pour les laboratoires à venir, il nous était difficile de percevoir quelles informations étaient primordiales. Ainsi, il n’était pas aisé de savoir si les identifiants seraient utiles.