



SCHÉMA JSON

SER – Laboratoire 3

Guidoux Vincent & Jaquet David

Table des matières

1	Petite introduction	2
2	Modification effectuée sur la structure XML	2
3	Objectif 1	2
3.1	Grammaire JSON Schema	2
3.2	Résultat de la validation	3
4	Objectif 2	3
4.1	Principe conceptuel de la solution	3
4.2	Code.....	4
4.3	Résultat obtenu sur navigateur web	5
5	Petite conclusion	7
6	Annexe.....	7
6.1	Fichiers.....	7

1 Petite introduction

Pour ce laboratoire, nous avons deux objectifs distincts ; le premier est de créer une grammaire JSON Schema permettant de valider le fichier JSON élaboré lors du précédent laboratoire.

Le second objectif est de créer un programme Java s'appuyant sur le fichier XML généré lors du précédent laboratoire. Le programme doit créer un nouveau fichier XML. Le nouveau fichier doit avoir un en-tête précis et doit pouvoir être validé par la DTD fournie.

2 Modification effectuée sur la structure XML

Nous nous sommes basés sur fichier XML fourni pour ce laboratoire afin de connaître les éléments qu'il nous manquait dans les fichiers que nous avons générés lors des précédents laboratoires. Nous avons donc rajouté la taille de la salle en attribut. L'objet film a également été modifié afin d'y ajouter l'attribut film_id. Les genres, les mots-clés, les acteurs ainsi que les langues ont un attribut no qui a également été rajouté. Il s'agit des seules modifications qui ont été apportées par rapport à l'XML du précédent laboratoire.

3 Objectif 1

3.1 Grammaire JSON Schema

Nous avons utilisé le JSON que nous avons générée lors du précédent laboratoire. Le JSON étant déjà correcte, nous n'avons pas effectué de changement sur ce dernier.

On peut voir dans la grammaire ci-contre que nous utilisons un tableau de projections. Chaque projection contient une date et un film.

La date est décomposée en sous élément représentant le jour, le mois, l'année, l'heure ainsi que les minutes de la projection. Chacun de ces éléments sont des entiers.

Le film est également séparé en sous éléments. Nous distinguons le titre, qui est du type string, ainsi qu'un tableau d'acteurs.

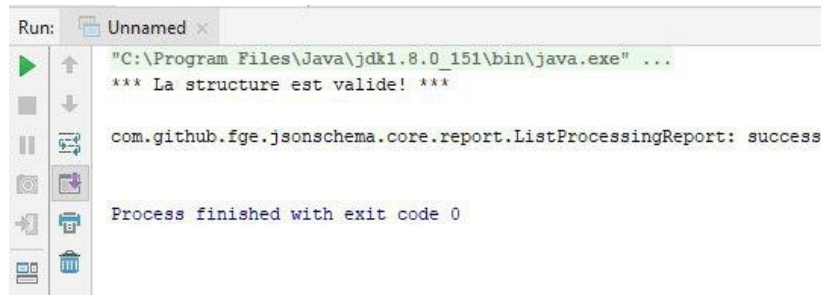
Les acteurs ne contiennent que leur nom de naissance. Ils sont toutefois des objets car les acteurs de la version JSON sont une version raccourcie par rapport aux acteurs de la version XML conformément à la donnée du laboratoire.

```
{
  "type": "object",
  "properties": {
    "projections": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "date": {
            "type": "object",
            "properties": {
              "jour": {"type": "integer"},
              "mois": {"type": "integer"},
              "annee": {"type": "integer"},
              "heure": {"type": "integer"},
              "minute": {"type": "integer"}
            }
          },
          "film": {
            "type": "object",
            "properties": {
              "titre": {"type": "string"},
              "acteurs": {
                "type": "array",
                "items": {
                  "type": "object",
                  "properties": {
                    "nom_naissance": {"type": "string"},
                    "role": {"type": "string"}
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

monschema.json

3.2 Résultat de la validation

Comme nous pouvons le voir sur l'image ci-dessous, notre grammaire ne nous génère pas d'erreur et est, par conséquent, valide. L'image est le résultat de l'exécution du programme `JSON_Validation` fourni pour ce laboratoire.



```
Run: Unnamed x
"C:\Program Files\Java\jdk1.8.0_151\bin\java.exe" ...
*** La structure est valide! ***

com.github.fge.jsonschema.core.report.ListProcessingReport: success

Process finished with exit code 0
```

4 Objectif 2

4.1 Principe conceptuel de la solution

Toute la solution réside dans un seul fichier java `Amorce.java` qui lit le fichier `plex.xml` (généré au laboratoire précédent), le parcourt en utilisant `Jdom2` et `jaxen`, prends les informations nécessaires, et recrée un fichier qui correspond à `projections.dtd`. Nous avons mis en évidence un exemple d'utilisation `XPath` pour aller chercher les informations sur les langues, le reste est exploré en `Jdom2`. Le fichier `.xsl` s'attend à un format « dd.MM.YYYY HH:mm » pour les heures, le notre étant moins épuré (une balise pour le jour, une pour le mois.. etc.) nous avons dû faire une petite adaptation dans la lecture du fichier. Nous n'avions pas fait une liste des acteurs dans notre fichier `.xml` du coup, il y a peut-être des doublons, pour éviter ça dans le fichier nécessaire à ce labo, nous avons fait une `ArrayList` d'acteurs déjà utilisés en sauvegardant leur `id`. Comme la note d'un film est une moyenne, cette dernière est considérée comme un nombre réel.

4.2 Code

Par soucis de place, nous avons décidé de n'afficher qu'une partie du code de la fonction `createXML` du fichier `Amorce.java`. Nous avons choisi de mettre dans ce rapport la partie contenant les projections ainsi que la partie contenant les langues afin d'avoir un exemple de `Jdom2` et de `XPath`.

```
private static void createXML(Document labo2) {

    try {

        Document document = new Document();
        XMLOutputter outp = new XMLOutputter(Format.getPrettyFormat());

        // -- Ajout d'une référence à la DTD
        DocType docType = new DocType("plex", "projections.dtd");
        document.addContent(docType);

        // -- Ajout de la référence à la feuille XSL
        ProcessingInstruction piXSL = new ProcessingInstruction("xml-stylesheet");
        HashMap<String, String> piAttributes = new HashMap<String, String>();
        piAttributes.put("type", "text/xsl");
        piAttributes.put("href", "projections.xsl");
        piXSL.setData(piAttributes);
        document.addContent(piXSL);

        //Element racine
        Element plex = new Element("plex");

        //-----
        //                               PROJECTIONS
        //-----
        Element projections = new Element("projections");

        for (Element projectionTMP : labo2.getRootElement().getChildren("projection")) {
            Element projection = new Element("projection");

            Element salle = new Element("salle");
            salle.setAttribute("taille",
projectionTMP.getChild("salle").getAttribute("taille").getValue());
            salle.addContent(projectionTMP.getChild("salle").getContent(0).getValue());

            projection.addContent(salle);

            Element date_heure = new Element("date_heure");
            Element date = projectionTMP.getChild("date");

            date_heure.setAttribute("format", "dd.MM.YYYY HH:mm");
            date_heure.addContent(getDateHour(date));

            projection.addContent(date_heure);

            projection.setAttribute("film_id",
projectionTMP.getChild("film").getAttributeValue("film_id"));
            projection.setAttribute("titre",
projectionTMP.getChild("film").getChild("titre").getValue());

            projections.addContent(projection);
        }

        plex.addContent(projections);
    }
}
```

```
//-----
//                                     LISTE_LANGAGES
//-----

Element liste_langages = new Element("liste_langages");

SAXBuilder builder = new SAXBuilder();
Document xmlDoc = builder.build(new File("plex.xml"));

XPathFactory xpfac = XPathFactory.instance();
XPathExpression xp = xpfac.compile("//film/langages/language", Filters.element());

List<Element> langages = (List<Element>) (xp.evaluate(xmlDoc));
for (Element l : langages) {
    Element language = new Element("language");
    language.addContent(l.getValue());
    language.setAttribute("no", l.getAttributeValue("no"));

    liste_langages.addContent(language);
}

plex.addContent(liste_langages);
```

4.3 Résultat obtenu sur navigateur web

Voici les résultats obtenus sur Firefox :

Plex - Projections

Film	Projection	Salle	Note
Tomorrow Never Dies	18.03.2018 17:09	Flon 1	4.0/5.0
Casino Royale	18.03.2018 22:08	Flon 2	1.0/5.0

Liste des projections

Plex - Projections



Tomorrow Never Dies



Durée : 119 minutes

Genres : Action - Crime - Adventure - Thriller

Langues : German - Cantonese - Mandarin - Danish - English

Synopsys

Agent 007, the unstoppable action hero must prevent a tremendous disaster ripped from tomorrow's headlines. Someone is pitting the world's superpowers against each other - and only James Bond can stop it. When a British warship is mysteriously destroyed in Chinese waters, the world teeters on the brink of World War III - until 007 zeros in on the true criminal mastermind. Bond's do-or-die mission takes him to Elliot Carver, a powerful media mogul who manipulates world events as easily as he changes headlines from his global media empire. After soliciting help from Carver's sexy wife, Paris, Bond join forces with a stunning yet lethal Chinese agent, Wai Lin, in a series of explosive chases, brutal confrontations and breathtaking escapes as they race to stop the presses on Carver's next planned news story: global pandemonium!

Critiques

3/5 : Le film était bien, mais mon ex était dans la salle

5/5 : Très bon film

Rôles

Place	Personnage	Acteur
1	James Bond	Brosnan, Pierce
2	Elliot Carver	Pryce, Jonathan
38	Staff Officer #2	Brattan, Laura
39	Beth Davidson	Cameron-Blakey, Nadia
40	Mary Golson	Ross, Liza
41	Jeff Hobbs	Napier, Hugo
42	Philip Jones	Saxon, Rolf
43	MiG Pilot	Wang, Vincent
44	General Chang	Kwok, Philip
9223372036854775807	Vietnamese Villager	Boo Khoo, Ian

Mots-cles

red-china	terrorism	ex-boyfriend-ex-girlfriend-relationship	terrorist	navy	sea	woman-slaps-man	blockbuster	red-chinese-agent	old-flame	1990s
heroin-addict	motorcycle	chakra	media-manipulation	hamburg-germany	secret-service	damsel-in-distress	mercenary	pistol	assassination-attempt	
asian-woman	woman-agent	vietnam	asian	husband-wife-relationship	london-england	air-base	tv-news	afghanistan	woman-spy	gadget-car
female-martial-artist	nuclear-weapons	exploding-helicopter	female-spy	machine-gun	exploding-ship	ship	cult-figure	female-agent	automobile	black-dress

Détail d'un film

Brosnan, Pierce



Sexe : MASCULIN

Nom de naissance : Brosnan, Pierce Brendan

Date de naissance : 16.04.1953

Date de décès :

Biographie

Pierce Brosnan was born in Navan, County Meath, Ireland, and lived in Navan, County Meath, until he moved to England, UK, at an early age (thus explaining his ability to play men from both backgrounds convincingly). His father left the household when Pierce was a child and although reunited later in life, the two have never had a close relationship. His most popular role is that of British secret agent James Bond. The death, in 1991, of 'Cassandra Harris (I)' (qv), his wife of eleven years, left him with three children - Christopher and Charlotte from Cassandra's first marriage and Sean from their marriage. Since her death, he has had two children with his second wife, 'Keely Shaye Smith' (qv).

Détail d'un acteur

5 Petite conclusion

Le premier objectif ne nous a pas posé de problème. Nous avons pu le terminer assez rapidement. Toutefois, le second objectif nous a pris beaucoup plus de temps que ce que nous avions prévu. Nous devons également modifier notre structure XML afin qu'il corresponde au mieux au résultat désiré. Mis à part cet imprévu temporel, nous n'avons pas eu de difficulté à réaliser ce second objectif.

6 Annexe

6.1 Fichiers

- monschema.json : JSONSchema
- plex.xml : XML source (du labo 2)
- projections.xml : XML généré (projections.xml)
- structure.json : JSON (du labo 2)
- Amorce.java : Fichier servant à la création du fichier `projection.xml`