

SER : LABORATOIRE #2

Johanna Melly & Yohann Meyer

Professeur
Eric Lefrançois

16 avril 2018

Table des matières

1	Introduction	2
2	Code	2
2.1	XML	2
2.2	JSON	2
3	Conclusion	2
4	Code XML	3
5	Code JSON	6

1 Introduction

L'objectif de ce laboratoire était de générer des fichiers XML et JSON à partir d'objets Java. Nous avons à disposition, pour nous faciliter la tâche, la librairie JDOM (pour l'XML) et la librairie GSON (pour le JSON). Pour le document JSON, nous avons dû reconstruire intégralement la structure du document. Pour le JSON, la génération du document se fait simplement en lui passant un objet, nous avons donc dû construire cet objet correctement.

2 Choix de structure

Nous avons pu garder la structure XML que nous avons choisie au premier laboratoire, car elle était implémentable sans trop de difficulté avec la structure des objets mise à disposition dans le projet Java fourni.

2.1 XML

Pour la génération du XML, nous avons simplement itéré sur les projections, puis créé ses éléments enfants. Pour chaque élément enfant, nous avons, si nécessaire, répété l'opération. Nous avons ajouté des tests là où les champs n'étaient pas obligatoires, afin d'éviter des `nullPointerExceptions`.

2.2 JSON

Pour la génération du fichier JSON, nous ne pouvions pas simplement appliquer la fonction `toJson()` sur la liste complète des projections, car nous devons présenter une partie seulement des informations des projections. Ainsi, nous avons créé trois nouvelles classes, une pour chaque objet qui comprend des éléments enfants. Ainsi, nous avons créé :

- `Acteurs_simplifies` qui possède uniquement des champs pour les deux premiers rôles
- `Film_simplifies` qui possède des acteurs simplifiés, mais aussi un titre
- `Projections_simplifiees` qui possède un film simplifié, mais aussi un id et une date

Nous avons ensuite, dans le `controleurMedia`, créé une liste de projections simplifiées, dont chaque projection simplifiée est construite à partir des véritables projections.

3 Conclusion

Dans ce laboratoire, nous avons pu apprendre à utiliser les librairies JDOM et GSON qui sont très pratiques pour générer les fichiers correspondants de manière rapide et efficace. La création du fichier XML était un peu fastidieuse car il y avait beaucoup d'éléments différents, qu'il était un peu long de chercher les champs dont les valeurs risquaient d'être nulles et que nous devions tester, et parce qu'il fallait être attentif à ajouter tous chaque élément fils à son parent et à la racine. La génération du fichier JSON était plus rapide, car il suffit de passer un objet à la fonction `toJson()` de l'objet `Gson`. Ce laboratoire était donc un peu long mais pas d'une grande difficulté, mais nous a appris à utiliser des outils qui nous seront utiles.

4 Code XML

```
public void createXML() {
    new Thread() {
        public void run() {
            mainGUI.setAcknowledgeMessage("Creation XML... WAIT");
            long currentTime = System.currentTimeMillis();
            try {
                globalData = ormAccess.GET_GLOBAL_DATA();

                mainGUI.setWarningMessage("Creation XML: WAITING...");

                // Reucperation des projections
                List<Projection> projections = globalData.getProjections();
                // Creation de l'element racine
                Element projectionsElements = new Element("Projections");
                // Creation du document XML avec Projections comme racine
                Document plex_doc = new Document(projectionsElements);

                for(Projection projection : projections){ // Parcours des projections

                    Element projectionElement = new Element("Projection");
                    projectionElement.setAttribute("Id", "projection" +
                        String.valueOf(projection.getId()));

                    // ----- FILM -----
                    Film film = projection.getFilm();
                    Element filmElement = new Element("Film");
                    filmElement.setAttribute("Id", String.valueOf(film.getId()));
                    // Ajout des elements qui composent un film
                    filmElement.addContent(new Element("titre").setText(film.getTitre()));
                    filmElement.addContent(new
                        Element("duree").setText(String.valueOf(film.getDuree())));
                    if(film.getPhoto() != null)
                        filmElement.addContent(new Element("photo").setAttribute("url",
                            film.getPhoto()));
                    filmElement.addContent(new
                        Element("synopsis").setText(film.getSynopsis()));

                    // ----- ELEMENT ROLES -----
                    Set<RoleActeur> roles = film.getRoles();

                    for(RoleActeur role : roles){
                        //++roleNumber;
                        Element roleElement = new Element("Role");
                        roleElement.setAttribute("Id", String.valueOf(role.getId()));

                        // ----- ELEMENT ACTEUR -----
                        Acteur acteur = role.getActeur();
                        Element acteurElement = new Element("Acteur");
                        acteurElement.setAttribute("Id",
                            "acteur"+String.valueOf(acteur.getId()));
                        acteurElement.setAttribute("sexe",
                            String.valueOf(acteur.getSexe()));
                        // Ajout des elements qui composent un acteur
                        acteurElement.addContent(new
                            Element("nom").setText(acteur.getNom()));
                        if(acteur.getDateDeces() != null){
                            String deces =
                                String.valueOf(acteur.getDateDeces().get(Calendar.DAY_OF_MONTH))
                                + "/" +
                                acteur.getDateDeces().get(Calendar.MONTH) + "/" +
                                acteur.getDateDeces().get(Calendar.YEAR);
                            acteurElement.addContent(new Element("dateDeces").setText(deces));
                        }
                    }
                }
            }
        }
    }
}
```

```

    if(acteur.getDateNaissance()!=null){
        String naissance =
            String.valueOf(acteur.getDateNaissance().get(Calendar.DAY_OF_MONTH))
            + "/" +
            acteur.getDateNaissance().get(Calendar.MONTH) + "/" +
            acteur.getDateNaissance().get(Calendar.YEAR);
        acteurElement.addContent(new
            Element("dateNaissance").setText(naissance));
    }

    if(acteur.getNomNaissance()!=null)
        acteurElement.addContent(new
            Element("nomNaissance").setText(acteur.getNomNaissance()));
    if(acteur.getBiographie()!=null)
        acteurElement.addContent(new
            Element("biographie").setText(acteur.getBiographie()));

    roleElement.addContent(acteurElement);
    roleElement.addContent(new
        Element("place").setText(String.valueOf(role.getPlace())));
    roleElement.addContent(new
        Element("personnage").setText(role.getPersonnage()));

    filmElement.addContent(roleElement);
}

// ----- ELEMENT MOTCLE -----
Set<Motcle> motscles = film.getMotscles();

for(Motcle motcle : motscles){
    Element motcleElement = new Element("Motcle");
    motcleElement.setAttribute("Id", String.valueOf(motcle.getId()));
    motcleElement.addContent(new
        Element("label").setText(motcle.getLabel()));

    filmElement.addContent(motcleElement);
}

// ----- ELEMENT LANGAGE -----

Set<Langage> langages = film.getLangages();

for(Langage langage : langages){
    Element langageElement = new Element("Langage");
    langageElement.setAttribute("Id", String.valueOf(langage.getId()));
    langageElement.addContent(new
        Element("Label").setText(langage.getLabel()));

    filmElement.addContent(langageElement);
}

// ----- ELEMENT CRITIQUE -----

Set<Critique> critiques = film.getCritiques();

if(critiques.size()!=0){
    for(Critique critique : critiques){
        Element critiqueElement = new Element("Critique");
        critiqueElement.setAttribute("Id",
            String.valueOf(critique.getId()));
        critiqueElement.setAttribute("Note",
            String.valueOf(critique.getNote()));
        critiqueElement.addContent(new
            Element("texte").setText(critique.getTexte()));
    }
}

```

```

        filmElement.addContent(critiqueElement);
    }
}

// ----- ELEMENT GENRE -----

Set<Genre> genres = film.getGenres();

for(Genre genre : genres){
    Element genreElement = new Element("Genre");
    genreElement.setAttribute("Id", String.valueOf(genre.getId()));
    genreElement.addContent(new
        Element("Label").setText(genre.getLabel()));

    filmElement.addContent(genreElement);
}

projectionElement.addContent(filmElement);

// ----- ELEMENT SALLE -----

Salle salle = projection.getSalle();
Element salleElement = new Element("Salle");
salleElement.setAttribute("Id", String.valueOf(salle.getId()));
salleElement.addContent(new Element("numero").setText(salle.getNo()));
salleElement.addContent(new
    Element("taille").setText(String.valueOf(salle.getTaille())));

projectionElement.addContent(salleElement);

// ----- ELEMENT DATE -----

Calendar dateHeure = projection.getDateHeure();
Element dateElement = new Element("Date");

dateElement.addContent(new
    Element("jour").setText(String.valueOf(dateHeure.get(Calendar.DAY_OF_MONTH))));
dateElement.addContent(new
    Element("mois").setText(String.valueOf(dateHeure.get(Calendar.MONTH))));
dateElement.addContent(new
    Element("annee").setText(String.valueOf(dateHeure.get(Calendar.YEAR))));

projectionElement.addContent(dateElement);

projectionsElements.addContent(projectionElement);
}
XMLOutputter xmlOutput = new XMLOutputter();

xmlOutput.setFormat(Format.getPrettyFormat());
// Ecriture dans un fichier
xmlOutput.output(plex_doc, new FileWriter("./Projections.XML"));

mainGUI.setAcknowledgeMessage("Creation XML... DONE!");
}
catch (Exception e){
    mainGUI.setErrorMessage("Construction XML impossible", e.toString());
}
}
}

```

```
    }.start();  
}
```

5 Code JSON

```
public void sendJSONToMedia() {  
    new Thread(){  
        public void run() {  
            mainGUI.setAcknowledgeMessage("Envoi JSON ... WAIT");  
            //long currentTime = System.currentTimeMillis();  
            try {  
                globalData = ormAccess.GET_GLOBAL_DATA();  
                // Creation d'un objet Gson formatee  
                Gson gson = new GsonBuilder().setPrettyPrinting().create();  
  
                // Creation d'une liste de projections simplifiees vide  
                List<Projections_simplifiees> projections = new ArrayList<>();  
                // Recuperation des projections completes  
                List<Projection> projectionsComplettes = globalData.getProjections();  
                for(Projection projectionComplete : projectionsComplettes){ // Parcours  
                    des projections completes  
                    // Creation d'une projections simplifiee a partir de la projection  
                    complete  
                    Projections_simplifiees projectionSimplifiee = new  
                        Projections_simplifiees(projectionComplete);  
                    // Ajout de la projection simplifiee a la liste de projections  
                    simplifiees  
                    projections.add(projectionSimplifiee);  
                }  
  
                Writer writer = new FileWriter("./Projections.json");  
  
                try{  
                    // Ecriture dans un fichier  
                    gson.toJson(projections, writer);  
                    writer.flush();  
                }catch(Exception e ){  
                    System.out.println(e.getMessage());  
                }  
  
                mainGUI.setAcknowledgeMessage("Envoi JSON: DONE");  
            }  
            catch (Exception e){  
                mainGUI.setErrorMessage("Construction JSON impossible", e.toString());  
            }  
        }  
    }.start();  
}
```
