

Labo 2

Sérialisation XML avec Java Jdom2
Sérialisation JSON avec Java
& Google/Gson

Vincent Guidoux et David Jaquet

Avril 2018

Table des matières

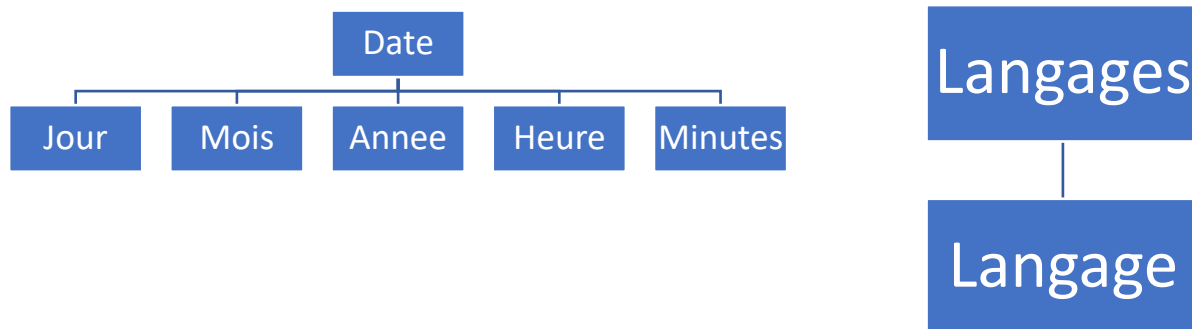
| | | |
|---|---|--|
| 1 | Petite introduction..... | 2 |
| 2 | Les modifications éventuelles apportées à vos structures XML et JSON élaborées dans le 1er laboratoire. Décrire alors l'objectif de ces modifications. | 2 |
| 3 | Présenter le code des contrôleurs ControleurXMLCreation et ControleurMedia | 3 |
| 4 | Commentez votre solution (méthodes rajoutées, complétées, leur objectif, points remarquables). | 5 |
| 5 | Présenter un extrait représentatif des deux fichiers que vous aurez générés. | Erreur ! Signet non défini. |
| 6 | Bilan - Conclusion | 7 |

1 Petite introduction

Nous avons pour objectif de générer les documents XML et JSON définis au laboratoire précédant tout en utilisant les librairies respectives Jdom2 et Google/Gson. Il sera intéressant de voir, une fois avoir interagi avec la base de données si nos arbres et graphes étaient correctes et bien pensés.

2 Modifications des structures

Nous avons oublié de mettre l'heure et les minutes pour les projections. Et nous avons aussi une autre idée que celle de la base de données concernant les langages



Pour la génération du fichier JSON Nous avons prévu un tableaux projections qui contenait un tableau d'objets projection. Mais maintenant, nous envoyons directement un tableau de projection.

3 Codes des contrôleurs ControleurXMLCreation et ControleurMedia

3.1 ControleurXMLCreation

```
Document doc = new Document();
XMLOutputter outp = new XMLOutputter(Format.getPrettyFormat());

Element projections = new Element("projections");

for (Projection p : globalData.getProjections()) {
    Element projection = new Element("projection");

    Element date = new Element("date");

    Calendar dateheure = p.getDateHeure();
    Element jour = new Element("jour");
    jour.addContent(Integer.toString(dateheure.get(Calendar.DAY_OF_MONTH)));
    Element mois = new Element("mois");
    mois.addContent(Integer.toString(dateheure.get(Calendar.MONTH)));
    Element annee = new Element("annee");
    annee.addContent(Integer.toString(dateheure.get(Calendar.YEAR)));
    Element heure = new Element("heure");
    heure.addContent(Integer.toString(dateheure.get(Calendar.HOUR_OF_DAY)));
    Element minute = new Element("minute");
    minute.addContent(Integer.toString(dateheure.get(Calendar.MINUTE)));

    date.addContent(jour);
    date.addContent(mois);
    date.addContent(annee);
    date.addContent(heure);
    date.addContent(minute);

    Element salle = new Element("salle");
    salle.addContent(p.getSalle().getNo());

    Element film = new Element("film");

    Film f = p.getFilm();

    Element titre = new Element("titre");
    titre.addContent(f.getTitre());
    Element synopsis = new Element("synopsis");
    synopsis.addContent(f.getSynopsis());
    Element duree = new Element("duree");
    duree.addContent(Integer.toString(f.getDuree()));

    Element critiques = new Element("critiques");
    for (Critique c : f.getCritiques()) {
        Element critique = new Element("critique");

        Element texte = new Element("texte");
        texte.addContent(c.getTexte());
        Element note = new Element("note");
        note.addContent(Integer.toString(c.getNote()));

        critique.addContent(texte);
        critique.addContent(note);

        critiques.addContent(critique);
    }

    Element genres = new Element("genres");
    for (Genre g : f.getGenres()) {
        Element genre = new Element("genre");
        genre.addContent(g.getLabel());

        genres.addContent(genre);
    }
}
```

```
}

Element mots_cles = new Element("mots_cles");
for (Motcle mc : f.getMotcles()) {
    Element mot_cle = new Element("mot_cle");
    mot_cle.addContent(mc.getLabel());

    mots_cles.addContent(mot_cle);
}

Element langages = new Element("langages");
for (Language l : f.getLangages()) {
    Element language = new Element("language");
    language.addContent(l.getLabel());

    langages.addContent(language);
}

Element photo = new Element("photo");
photo.setAttribute("url", "" + f.getPhoto());

Set<Acteur> acteurSet = new HashSet<>();

for (RoleActeur ra : f.getRoles()) {
    acteurSet.add(ra.getActeur());
}

[... ..]

    acteur.addContent(nom);
    acteur.addContent(nom_naissance);
    acteur.addContent(biographie);
    acteur.addContent(roles);
    acteur.addContent(date_naissance);
    acteur.addContent(date_deces);

    acteurs.addContent(acteur);
}

film.addContent(titre);
film.addContent(synopsis);
film.addContent(duree);
film.addContent(critiques);
film.addContent(genres);
film.addContent(mots_cles);
film.addContent(langages);
film.addContent(photo);
film.addContent(acteurs);

projection.addContent(date);
projection.addContent(salle);
projection.addContent(film);

projections.addContent(projection);
}

doc.addContent(projections);

DocType docType = new DocType("projections", "Plex.dtd");
doc.setDocType(docType);
outp.output(doc, new FileOutputStream("plex.xml"));

mainGUI.setAcknowledgeMessage("Creation XML... FINISHED");
```

3.2 ControleurMedia

```
globalData = ormAccess.GET_GLOBAL_DATA();

Gson moteurJson = new GsonBuilder().setPrettyPrinting().create();

BufferedWriter out = new BufferedWriter(new OutputStreamWriter(new
FileOutputStream("plex.json"), "UTF-8"));
JsonObject data = new JsonObject();
JsonArray projectionsList = new JsonArray();

for (Projection proj : globalData.getProjections()){
    JsonObject projection = new JsonObject();

    // Date
    JsonObject date = new JsonObject();
    Calendar projectionDate = proj.getDateHeure();
    date.add("jour", new JsonPrimitive(projectionDate.get(Calendar.DAY_OF_MONTH)));
    date.add("mois", new JsonPrimitive(projectionDate.get(Calendar.MONTH)));
    date.add("annee", new JsonPrimitive(projectionDate.get(Calendar.YEAR)));
    date.add("heure", new JsonPrimitive(projectionDate.get(Calendar.HOUR_OF_DAY)));
    date.add("minute", new JsonPrimitive(projectionDate.get(Calendar.MINUTE)));
    projection.add("date", date);

    // Film
    JsonObject film = new JsonObject();
    film.add("titre", new JsonPrimitive(proj.getFilm().getTitre()));

    // Acteur
    JsonArray actorList = new JsonArray();
    for (RoleActeur role : proj.getFilm().getRoles()){
        if (role.getPlace() == 1 || role.getPlace() == 2){
            JsonObject actor = new JsonObject();
            if (role.getActeur().getNomNaissance() != null)
                actor.add("nom_naissance", new
JsonPrimitive(role.getActeur().getNomNaissance()));
            else
                actor.add("nom", new JsonPrimitive(role.getActeur().getNom()));

            actorList.add(actor);
        }
    }

    film.add("acteurs", actorList);
    projection.add("film", film);

    projectionsList.add(projection);
}

data.add("projections", projectionsList);

out.write(moteurJson.toJson(data));
out.close();

mainGUI.setAcknowledgeMessage("Envoi JSON ... FINISHED");
```

4 Commentaires

4.1 XML

Nous avons implémenté la méthode `createXML()` afin qu'elle écrive dans un fichier `xml` les projections voulues suivant la définition de notre graphe du laboratoire 1. Quelques difficultés ont été rencontrées pour afficher les différents rôles de chaque acteur. Comme toute la base de données n'est pas chargée dans la variable `GlobalData`, il n'était pas possible d'utiliser la méthode `getRoles()` de la classe `Acteur`. Nous avons initialisé l'objet `XMLOutputter` avec `Format.getPrettyFormat` pour avoir une belle indentation dans le fichier. Pour remplir le fichier nous avons utilisé la librairie `Jdom2`, pratique et facile d'utilisation.

4.2 JSON

Pour la partie `JSON`, nous avons implémenté la méthode `sendJSONToMedia()` de la classe `ControleurMedia` afin qu'elle enregistre la liste des projections dans un fichier `JSON`. Comme pour le laboratoire précédent, nous avons fait une version raccourcie par rapport au `XML`. De cette manière, nous avons pu bien analyser les différences comme décrit au point 2.

Nous utilisons l'objet `GsonBuilder` avec la méthode `setPrettyPrinting()` afin d'avoir un résultat agréable à la lecture.

Nous parcourons toutes les projections que nous avons et, à chaque itération, nous créons un objet `date` avec tous ses attributs, un objet `film` qui contient le titre ainsi ses acteurs principaux avec leur nom de naissance. Il est important de noter que le nom de naissance n'est pas rempli pour tous les acteurs. C'est pourquoi nous avons décidé l'attribut `nom de l'acteur` si le `nom de naissance` était vide.

5 Fichiers générés

Voici un extrait des fichiers que nous avons générés (JSON à gauche, XML à droite) :

```
{
  "projections": [
    {
      "date": {
        "jour": 19,
        "mois": 4,
        "annee": 2018,
        "heure": 10,
        "minute": 52
      },
      "film": {
        "titre": "Tomorrow Never Dies",
        "acteurs": [
          {
            "nom_naissance": "Brosnan, Pierce Brendan"
          },
          {
            "nom_naissance": "Price, John"
          }
        ]
      }
    },
    {
      "date": {
        "jour": 19,
        "mois": 3,
        "annee": 2018,
        "heure": 10,
        "minute": 52
      },
      "film": {
        "titre": "Inglourious Basterds",
        "acteurs": [
          {
            "nom": "Laurent, Mélanie"
          },
          {
            "nom_naissance": "Pitt, William Bradley"
          }
        ]
      }
    }
  ]
}
```

plex.json

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE projections SYSTEM "Plex.dtd">
<projections>
  <projection>
    <date>
      <jour>19</jour>
      <mois>4</mois>
      <annee>2018</annee>
      <heure>10</heure>
      <minute>52</minute>
    </date>
    <salle>Flon 1</salle>
    <film>
      <titre>Tomorrow Never Dies</titre>
      <synopsis>Agent 007, [...]</synopsis>
      <duree>119</duree>
      <critiques>
        <critique>
          <texte>Je me suis trompé de salle</texte>
          <note>3</note>
        </critique>
      </critiques>
      <genres>
        <genre>Action</genre>
      </genres>
      <mots_cles>
        <mot_cle>saigon-vietnam</mot_cle>
      </mots_cles>
      <langages>
        <langage>English</langage>
      </langages>
      <photo_url>"null" />
      <acteurs>
        <acteur sexe="MASCULIN">
          <nom>Matthews, Al</nom>
          <nom_naissance />
          <biographie>[...]</biographie>
          <roles>
            <role>
              <nom>Master Sergeant 3</nom>
              <place>20</place>
            </role>
          </roles>
          <date_naissance>
            <jour>21</jour>
            <mois>10</mois>
            <annee>1942</annee>
          </date_naissance>
          <date_deces>
            <jour />
            <mois />
            <annee />
          </date_deces>
        </acteur>
      </acteurs>
    </film>
  </projection>
</projections>
```

plex.xml

6 Petite conclusion

Nous sommes prêts pour le laboratoire 3, notre DTD prends le fichier qui a été généré et arrive à le parser. Nous sommes satisfait du résultat de ce laboratoire et nous pensons que les objectifs qui ont été annoncés en introduction ont été remplis.