

Projet XML

- Thomas Farineau
- Léo Kitabdjian

Table des matières

1. Schéma XML
 - Avantages de notre modélisation
 - Inconvénients de notre modélisation
2. Scénarios implémentés avec XSLT
 - Scénario 1
 - Scénario 2
 - Scénario 3
3. Scénario avec une nouvelle structure XML
4. Scénario avec une nouvelle structure JSON
5. Outil python de compilation XSL et de serveur web
 - Installation
 - Utilisation

Schéma XML

Le schéma XML est disponible dans le fichier [/schema/schema.xsd](#).

Le schéma XML décrit la structure de notre organisme proposant des séjours linguistiques. Le schéma définit plusieurs types simples et un élément principal, “organisme”, qui contient des éléments enfants pour décrire les clients, les accompagnateurs, les enseignants et les séjours.

Voici les types simples définis dans le schéma :

1. typePublic : un type pour représenter le public cible des séjours, avec trois valeurs possibles - adulte, famille et enfant-adolescent.
2. typeActivite : un type pour représenter les types d’activités proposées pendant les séjours, avec trois valeurs possibles - CULTURE, SPORT et LOISIR.
3. niveau : un type pour représenter le niveau des cours de langue, avec trois valeurs possibles - DEBUTANT, INTERMEDIAIRE et AVANCE.
4. test : un type pour indiquer si un test est requis pour un cours, avec deux valeurs possibles - TEST et PAS DE TEST.
5. typeSejour : un type pour représenter le type de séjour linguistique, avec deux valeurs possibles - immersion_en_famille et stage-linguistique.
6. telephone : un type pour valider le format des numéros de téléphone.
7. email : un type pour valider le format des adresses e-mail.
8. langue : un type pour représenter les langues disponibles. (anglais, français, italien, espagnol, allemand)

L'élément "organisme" est composé des éléments suivants :

1. clients : une liste de clients, chacun ayant un ID, un nom, un prénom, un e-mail, un numéro de téléphone et un type de public.
2. accompagnateurs : une liste d'accompagnateurs, chacun ayant un ID, un nom, un prénom, un e-mail et un numéro de téléphone.
3. enseignants : une liste d'enseignants, chacun ayant un ID, un nom, un prénom, un e-mail, un numéro de téléphone et une liste de langues qu'ils enseignent.
4. séjours : une liste de séjours linguistiques, chacun ayant un ID, un type de séjour, une destination, un nombre de places, des dates de début et de fin, un prix, une liste de publics cibles, une liste de langues, un accompagnateur, un enseignant, une liste d'activités et une liste de cours.

Chaque activité a une description, un type d'activité et un prix, tandis que chaque cours a un nom, un niveau, une langue et un indicateur de test.

Avantages de notre modélisation

- Structuration claire : La modélisation organise clairement les informations et relations entre les éléments (clients, accompagnateurs, enseignants, séjours, etc.), facilitant la compréhension et la gestion des données.
- Validation des données : La modélisation utilise des types simples et des restrictions pour s'assurer que les données respectent un format spécifique. Par exemple, les types "email" et "telephone" définissent des modèles pour valider les adresses e-mail et les numéros de téléphone.
- Extensibilité : La modélisation peut être facilement étendue pour inclure de nouveaux types ou éléments, ce qui permet d'évoluer avec les besoins de l'application.
- Réutilisation : Les types simples définis dans le schéma peuvent être réutilisés pour d'autres éléments ou attributs si nécessaire, ce qui favorise la cohérence et la modularité.

Inconvénients de notre modélisation

1. Verbose : Le langage XML est souvent verbeux, ce qui peut rendre la lecture et la rédaction des données plus difficile et moins intuitive pour les humains.
2. Taille des fichiers : La modélisation XML peut entraîner des fichiers de taille plus importante en raison de la répétition des balises et des attributs.
3. Complexité : La modélisation XML peut être complexe, surtout pour les schémas plus volumineux. Cela peut entraîner une courbe d'apprentissage plus raide pour les développeurs qui ne sont pas familiers avec XML.
4. Performance : L'analyse et la manipulation des documents XML peuvent être plus lentes que d'autres formats de données, comme JSON, en raison de la nature hiérarchique et verbeuse du langage.

Notre modélisation présente des avantages en termes de structuration et de validation des données, mais présente également des inconvénients liés à la verbosité, la taille des fichiers, la complexité et la performance.

Scénarios implémentés avec XSLT

Scénario 1

Description : Afficher des statistiques sur l'organisme.

Intérêt : Ce scénario permet de connaître un grand nombre de statistiques sur les clients, enseignants et séjours, permettant de donner une vue d'ensemble plus complète et chiffrée aux administrateurs de l'organisme.

Entrée : </scenarios/statistiques/statistiques.xsl>

Sortie : </scenarios/statistiques/statistiques.html>

CSS Associé : </style/public.css>

Scénario 2

Description : Afficher les séjours détaillés.

Intérêt : Ce scénario permet de donner une vue d'ensemble claire et détaillée sur l'ensemble des séjours proposés. Ces derniers sont classés par type et regroupent toutes les informations les concernant. Ainsi un personnel de l'organisme peut accéder aisément aux informations détaillées des séjours.

Entrée : </scenarios/sejours/sejours.xsl>

Sortie : </scenarios/sejours/sejours.html>

CSS Associé : </style/sejours.css>

Scénario 3

Description : Afficher les potentiels professeurs affectables aux séjours.

Intérêt : Ce scénario permet d'afficher pour chaque séjour et pour chaque langue de ces séjours la liste des professeurs pouvant remplir ce rôle. De plus il indique si le professeur est déjà affecté à ce séjour ou non.

Entrée : </scenarios/affectation/affectation.xsl>

Sortie : </scenarios/affectation/affectation.html>

CSS Associé : </style/affectation.css>

Scénario avec une nouvelle structure XML

Description : Nous souhaitons extraire uniquement les informations sur les séjours linguistiques, les clients et les enseignants. Ce scénario permet de structurer les données différemment pour les intégrer dans un autre système (scolaire pour suivre notre idée).

Entrée : /scenarios/new_structure/new_structure.xsl

Sortie : /scenarios/new_structure/new_structure.xml

Scénario avec une nouvelle structure JSON

Description : Nous souhaitons extraire uniquement les informations sur les clients et les accompagnateurs pour une application mobile qui permet de visualiser la liste des participants aux séjours linguistiques et leurs informations de contact.

Problème rencontré : La bibliothèque lxml ne prend pas encore en charge la version XSLT 3.0. Par conséquent, nous avons dû modifier notre approche et trouver une solution pour utiliser une version compatible avec lxml afin que notre code Python puisse générer le fichier JSON.

Entrée : </scenarios/json/json.xsl>

Sortie : </scenarios/json/json.json>

Outil python de compilation XSL et de serveur web

Installation

Prérequis

- Python 3.6 ou supérieur
- pip

Installation des dépendances

Pour installer les dépendances, il suffit de lancer la commande suivante :

```
pip install -r requirements.txt
```

Utilisation

```
py app.py
```

Un serveur web s'ouvre sur le port 8000 et 3 url sont accessibles :

- <http://localhost:8000/statistiques>
- <http://localhost:8000/sejours>
- <http://localhost:8000/affectation>

Ici chaque fichier est généré à partir d'un fichier XSLT et d'un fichier XML de manière dynamique à chaque rechargement de la page.