



Compte Rendu du Projet XML

Réaliser par :

Houssem AYOUB

Younes HADDAM

À l'attention de :

Etienne REITH

Stéphane HÉRAUVILLE

Table des matières

I.	Introduction	3
II.	Accès au serveur Clever Cloud.....	3
III.	Les chemins définis dans le service REST	3
IV.	Choix technique	7
IV.1	Persistance	7
IV.2	Sérialisation/désérialisation des flux XML	7
IV.3	Génération des pages HTML	8
IV.4	Gestion des erreurs.....	8
V.	Architecture du projet.....	8
V.1	Structure des paquets	8
V.2	Structure de la base de données.....	8
VI.	Manuelle d'installation et d'utilisation	9
VI.1	Installation du projet	9
VI.2	Utilisation du service	10
VII.	Conclusion.....	15

I. Introduction

Ce projet consiste à créer un service web, permet de gérer un flux XML¹. Il fournit une API ²REST ³pour insérer, lister et supprimer des données représentées sous ce format. Les données insérées seront persistées dans une base de données. Ce service permet aussi de transformer ce flux en un flux HTML⁴.

II. Accès au serveur Clever Cloud

Nom d'utilisateur : houssem.ayoub@univ-rouen.fr

Mot de passe : Houssem_833

III. Les chemins définis dans le service REST

- **Page d'accueil**

Chemin : /

Méthode : GET

Format de retour : HTML

Affiche la page d'accueil du service.

- **Page d'aide**

Chemin : /help

Méthode : GET

Format de retour : HTML

Affiche une page qui décrit les chemins fournis par le service, la méthode utilisée pour communiquer avec l'URL⁵ et la structure de données retourné par ce dernier.

¹ eXtensible Markup Language (langage de balisage extensible)

² application programming interface (interface de programmation d'application)

³ representational state transfer

⁴ HyperText Markup Language (langage de balises pour l'hypertexte)

⁵ Uniform Resource Locator

- **Liste des articles**

Chemin : /rss22/resume/xml

Méthode : GET

Format de retour : XML

Retourne un flux XML contenant l'identifiant, le titre, et la date de publication ou de mise à jour de tous les articles présents dans la base de données du service.

```
<items>
```

```
  <item guid= {guid}>
```

```
    <title>{titre}</title>
```

```
    <date>{date}</date>
```

```
  </item>
```

```
  ...
```

```
</items>
```

Chemin : /rss22/resume/html

Méthode : GET

Format de retour : HTML

Retourne une page web contenant un tableau qui liste l'identifiant, le titre, la date de publication ou de mise à jour de tous les articles présents dans la base de données du service.

- **Détails d'un article**

Chemin : /rss22/xml/{guid}

Méthode : GET

Format de retour : XML

Retourne un flux XML contenant les détails d'un article, dont son identifiant a été fournis par l'URL (guid), en respectant la structure du schéma rss22.xsd définit au cours du TP.

Chemin : /rss22/html/{guid}

Méthode : GET

Format de retour : HTML

Retourne une flux HTML qui présente les détails d'un article, dont son identifiant a été fournis par l'URL (guid).

- **Ajout d'un article dans la base**

Chemin : /rss22/insert

Méthode : POST

Format de retour : XML

Transmis : Un flux XML à jouter, conforme au schéma rss22.xsd

Le flux reçu est validé par le schéma XSD ⁶et ajouté à la base de données.

Dans le cas où l'opération est réussite un flux xml sera retourné sous la forme :

<response>

 <guid>{identifiant de dernier article ajouté } </guid>

 <status>INSERTED</status>

 <description>Le flux est ajouté avec succès</description>

</response>

⁶ XML Schema Definition

En cas d'erreur de validation ou que le flux existe déjà dans la base (même titre et même date) un flux xml sera retourné sous la forme :

```
<response>

    <status>ERROR</status>

    <description>{une description d'erreur}</description>

</response>
```

- **Suppression d'un article**

Chemin : /rss22/delete/{guid}

Méthode : DELET

Format de retour : XML

L'article avec l'identifiant {guid} sera supprimé. Si le flux enregistré dans la base de données n'est plus attaché à aucun article, il sera supprimé.

Dans le cas où l'opération est réussite un flux xml sera retourné sous la forme :

```
<response>

    <guid>{identifiant de dernier article ajouté } </guid>

    <status>DELETED</status>

    <description>L'article est supprimé avec succès</description>

</response>
```

En cas d'erreur un flux xml sera retourné sous la forme :

```
<response>

    <status>ERROR</status>

    <description>{une description de l'erreur}</description>

</response>
```

- **API de recherche**

Chemin : /rss22/resume/search?...

Méthode : GET

Format de retour : XML

Retourne un flux XML contenant l'identifiant, le titre, et la date des articles présents dans la base de données et conforme aux critères de la recherche.

Les paramètres possibles pour la recherche :

date (sous la forme [YYYY]) :

Sélectionne les articles, qui ont l'année égal ou postérieur à l'année donnée au paramètre.

titreListe (sous forme m1 ;m2 ;...):

Sélectionne les articles, dont leurs titres un des mots séparés par un point-virgule.

IV. Choix technique

IV.1 Persistance

Dans ce projet on a choisi d'utiliser le cadriciel « Hibernate » pour le système de gestion de la base de données MYSQL⁷. On a fait ce choix, car les données qui vont être stocké respecte la même structure de données, et MYSQL est le plus fiable pour gérer des données structurées. Il fournit une meilleure performance d'accès et de recherche à travers son système d'indexation.

IV.2 Sérialisation/désérialisation des flux XML

La sérialisation/désérialisation du flux XML se fait avec le cadriciel JAXB⁸. Il permet, à travers des class annotées, d'automatiser l'opération et d'avoir plus de flexibilité concernant la forme du flux généré.

⁷ Un système de gestion de bases de données relationnelles

⁸ Java Architecture for XML Binding

IV.3 Génération des pages HTML

La génération des pages HTML (liste des articles, et détails d'article) se font par, la transformation du flux XML, généré à partir des objets récupérés de la base de données, à l'aide des fichiers XSLT⁹.

IV.4 Gestion des erreurs

La gestion des erreurs se fait, à l'aide de deux contrôleurs. Un contrôle pour la gestion du flux HTML et l'autre pour le flux XML, qui captent les exceptions déclencher par les méthodes et qui génèrent un retour adéquat.

V. Architecture du projet

V.1 Structure des paquets

Controller : Contient les contrôleurs qui gèrent les chemins du service et les écouteurs des exceptions.

Exception : Contient les classes d'exceptions définit dans le projet

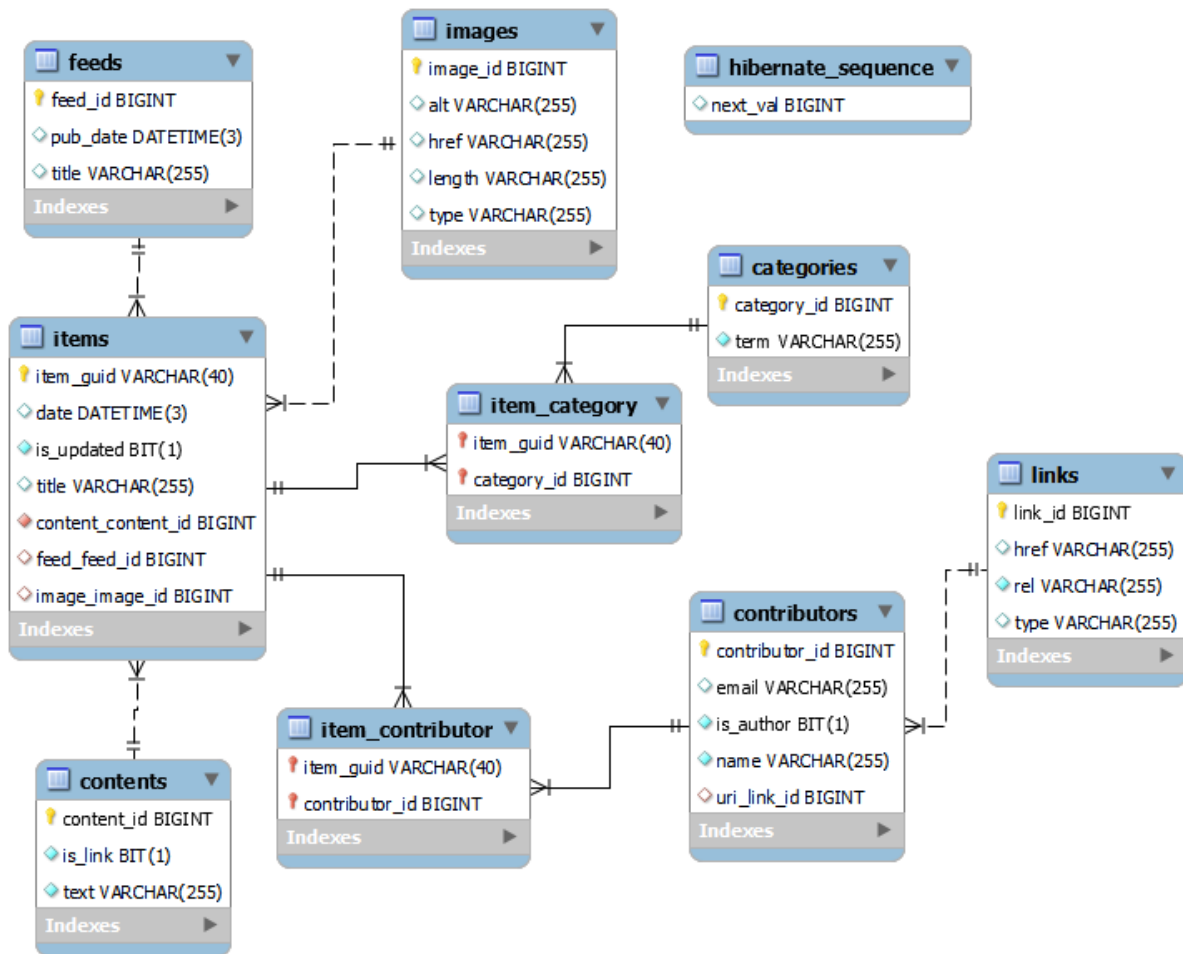
Model : Contient les modèles des données XML, ORM¹⁰, le validateur XML, les transformateurs, et adaptateurs des objets. Il contient aussi, les services permettent de communiquer avec la base de données.

V.2 Structure de la base de données

Le diagramme suivant représente la structure de la base de données et les relations entre les tables.

⁹ Extensible Stylesheet Language Transformation

¹⁰ Object-relational mapping



VI. Manuelle d'installation et d'utilisation

VI.1 Installation du projet

Pour récupérer le projet en local, il faut le cloner à partir du github.

```
$ git clone https://github.com/housseem-ayoub/rss22.git
```

Le service a besoin d'une variable d'environnement qui définit L'URL de la base de données.

```
MYSQL_ADDON_URI : mysql://{nom d'utilisateur}:{mot de
passe}@{serveur}:{port}/{nom de la base de données}
```

Pour démarrer le service, il faut exécuter la commande suivante dans le répertoire du projet.

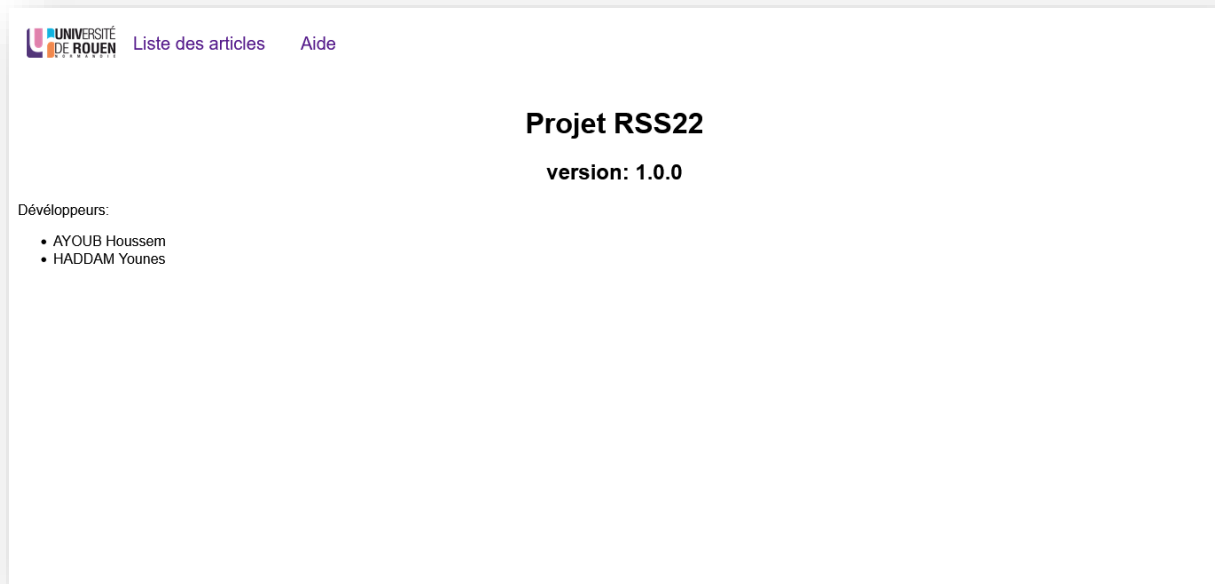
```
$ ./mvnw spring-boot :run
```

VI.2 Utilisation du service

On peut accéder aux URLs qui retournent un flux HTML à travers un navigateur web.

Page d'accueil :

URL : <https://rss22-ayoub-haddam.cleverapps.io/>




Page d'aide :

URL : <https://rss22-ayoub-haddam.cleverapps.io/help>

Liste des requêtes:		
URI	Methode	Résumé de l'Opération
/	GET	Affiche la page d'accueil sous format HTML
/help	GET	Affiche la page contenant les informations d'aide
/rss22 /resume/xml	GET	Affiche la liste des articles stockés sous format XML Retour: <items> <item> <guid>{guid}</guid> <title>{title}</title> <date>{date}</date> </item> ... </items>
/rss22 /resume/html	GET	Affiche la liste des articles stockés sous format HTML
/rss22 /xml/{guid}	GET	Affiche le contenu complet de l'article dont l'identifiant est {guid} sous Format XML
/rss22 /html/{guid}	GET	Affiche le contenu complet de l'article dont l'identifiant est {guid} sous Format HTML
rss22/insert	POST	stocker Les articles du flux dans la base de données. Transmis: un flux XML qui respecte le schéma rss22.xsd Retour: Un flux xml sous la forme : <response> <guid>{guid}</guid> <status>INSERTED</status> <description>{message}</description>



Liste des articles :

URL : <https://rss22-ayoub-haddam.cleverapps.io/rss22/resume/html>

 Accueil Aide			
List des articles			
Sommaire:	GUID	TITRE	DATE
n°			
1	0a5a22ee-aab8-4880-a197-bbf62e7aa807	Element 01	22-02-2022 20:20:10
2	f613f52b-1477-4f8e-a17b-0f9a6e14b206	Element 02 qui depasse les 45 caracteres et q	10-02-2022 20:20:10

Détails d'un article :

URL : <https://rss22-ayoub-haddam.cleverapps.io/rss22/html/0a5a22ee-aab8-4880-a197-bbf62e7aa807>

 Accueil Liste des articles Aide			
Element 01			
Catégories: cat2 cat1			
publié le 22-02-2022 20:20:10			
			
http://www.corp.gov/sceptrat/et			

Si l'identifiant est incorrecte une page d'erreur s'affiche

La page demandé n'existe page

GUID: 0a5a22ee-aab8-4880-a197-bbf62e7aa804

Status: ERROR

On peut interagir avec les URLs qui retourne un flux XML en utilisant POSTMAN.

Liste de articles :

URL : <https://rss22-ayoub-haddam.cleverapps.io/rss22/resume/xml>

Méthode : GET

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rss:items xmlns:rss="http://univrouen.fr/rss22">
  <rss:item guid="f613f52b-1477-4f8e-a17b-0f9a6e14b206">
    <rss:title>Element 02 qui depasse les 45 caracteres et qui va etre coupe</rss:title>
    <rss:date>2022-02-10T20:20:10.010Z</rss:date>
  </rss:item>
  <rss:item guid="0a5a22ee-aab8-4880-a197-bbf62e7aa807">
    <rss:title>Element 01</rss:title>
    <rss:date>2022-02-22T20:20:10.010Z</rss:date>
  </rss:item>
</rss:items>
```

Détails d'un article :

URL : <https://rss22-ayoub-haddam.cleverapps.io/rss22/xml/0a5a22ee-aab8-4880-a197-bbf62e7aa807>

Méthode : GET

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rss:item xmlns:rss="http://univrouen.fr/rss22" guid="0a5a22ee-aab8-4880-a197-bbf62e7aa807">
  <rss:title>Element 01</rss:title>
  <rss:category term="cat2"/>
  <rss:category term="cat1"/>
  <rss:published>2022-02-22T20:20:10.010Z</rss:published>
  <rss:image type="PNG" href="https://picsum.photos/500/300" alt="image du flux" length="3"/>
  <rss:content type="src" href="http://www.corp.gov/sceptra/et"/>
  <rss:author>
    <rss:name>Anthony Kasper</rss:name>
    <rss:email>t@t.fr</rss:email>
    <rss:uri rel="self" type="text/html" href="http://www.any.org/speluncis/profundum"/>
  </rss:author>
  <rss:author>
    <rss:name>Sophy Jose</rss:name>
  </rss:author>
</rss:item>
```

Si l'identifiant est incorrect un flux XML d'erreur sera retourné.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rss:response xmlns:rss="http://univrouen.fr/rss22">
  <rss:guid>0a5a22ee-aab8-4880-a197-bbf62e7aa806</rss:guid>
  <rss:status>ERROR</rss:status>
  <rss:description>l'article demandé n'existe pas</rss:description>
</rss:response>
```

API de recherche :

URL : <https://rss22-ayoub-haddam.cleverapps.io/rss22/search?date=2015&titreListe=article;depasse>

Méthode : GET

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rss:items xmlns:rss="http://univrouen.fr/rss22">
  <rss:item guid="f4d37064-4ccb-447e-8790-f8812b17942c">
    <rss:title>Mon Article</rss:title>
    <rss:date>2015-02-22T20:20:10.010Z</rss:date>
  </rss:item>
  <rss:item guid="f613f52b-1477-4f8e-a17b-0f9a6e14b206">
    <rss:title>Element 02 qui depasse les 45 caracteres et qui va etre coupe</rss:title>
    <rss:date>2022-02-10T20:20:10.010Z</rss:date>
  </rss:item>
</rss:items>
```

Insertion d'un flux XML :

URL : <https://rss22-ayoub-haddam.cleverapps.io/rss22/insert>

Méthode : POST

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rss:response xmlns:rss="http://univrouen.fr/rss22">
  <rss:guid>d986b9bd-2068-4416-b2c8-3128996b1a49</rss:guid>
  <rss:status>INSERTED</rss:status>
  <rss:description>Le flux a été ajouté avec succès</rss:description>
</rss:response>
```

Si le flux est invalide une réponse d'erreur sera retourné.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rss:response xmlns:rss="http://univrouen.fr/rss22">
  <rss:status>Error</rss:status>
  <rss:description>cvc-complex-type.2.4.b: The content of element 'rss:contributor' is not complete. One of '
    {"http://univrouen.fr/rss22":name, "http://univrouen.fr/rss22":email, "http://univrouen.fr/rss22":uri}' is
    expected.</rss:description>
</rss:response>
```

Aussi quand le flux existe déjà.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rss:response xmlns:rss="http://univrouen.fr/rss22">
  <rss:status>Error</rss:status>
  <rss:description>Le flux existe déjà</rss:description>
</rss:response>
```

Suppression d'un article :

URL : <https://rss22-ayoub-haddam.cleverapps.io/rss22/delete/b793deb5-379a-4426-af81-f1e9299eef79>

Méthode : DELETE

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rss:response xmlns:rss="http://univrouen.fr/rss22">
  <rss:guid>b793deb5-379a-4426-af81-f1e9299eef79</rss:guid>
  <rss:status>DELETED</rss:status>
  <rss:description>L'article a été supprimé avec succès</rss:description>
</rss:response>
```

Si l'identifiant n'existe pas une réponse d'erreur sera retourné.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rss:response xmlns:rss="http://univrouen.fr/rss22">
  <rss:guid>b793deb5-379a-4426-af81-f1e9299eef79</rss:guid>
  <rss:status>ERROR</rss:status>
  <rss:description>l'article demandé n'existe pas</rss:description>
</rss:response>
```

VII. Conclusion

Ce projet nous a permis de comprendre le fonctionnement du Spring boot, et la façon de créer un service REST à partir de ce cadriciel. IL nous a aussi permis, de maîtriser la structure XML, sa validation à partir d'une définition XSD, et sa transformation à travers d'un fichier XSLT. Dans ce projet on a appris L'utilisation d'Hibernate, et son intégration avec Spring boot. Il nous a été, le billet de maitriser des technologies demandées dans le Domain professionnel dont on va l'intégrer dans le futur proche.