



Projet Langage Web 2

XML

Réalisé par :

TOUBAL Zine-Eddine G3
BEN AHMED Yacine G1

Encadré par :

Mr E. REITH
Mr S. HÉRAUVILLE

Année Universitaire 2023/2024

Table des matières

1	Introduction	1
	Liste des acronymes	3
2	Configuration requise	4
2.1	Application déployée	4
2.2	Application déployée en local	4
3	Conception	5
3.1	Model	5
3.1.1	Architecture	5
3.1.2	Description des fichiers	6
3.2	Controllors	9
3.3	Repositories	9
3.4	Util	9
3.5	Exceptions	10
3.6	Mapper	10
3.7	Services	10
4	Déploiement	11
4.1	Requête réalisée	11
4.1.1	Insertion	11
4.1.2	Page help	12
4.1.3	Page d'accueil	13
4.1.4	Page d'erreur	14
4.1.5	Liste des CV format XML	15
4.1.6	Liste des CV format HTML	16
4.1.7	CV format XML	17
4.1.8	CV format HTML	18
4.1.9	Suppression d'un CV	18
4.1.10	Message d'erreur	19
5	Références	22

Table des figures

1	Configuration de la base de données	4
2	Packages	5
3	Diagramme de classe	6
4	Configuration de la base de données	11
5	Insertion d'un flux XML	11
6	Insertion d'un flux XML non valide	12
7	Page d'aide	13

8	Page d'accueil	14
9	Page d'erreur	15
10	Liste des CV format XML	16
11	Liste des CV format HTML	17
12	Détail d'un CV format XML	17
13	Détail d'un CV format HTML	18
14	Suppression d'un CV	19
15	Erreur flux déjà existant	19
16	Suppression CV inconnu	20
17	Affichage format XML d'un CV inconnu	20
18	Affichage format HTML d'un CV inconnu	20
19	Affichage Liste des CV format HTML inconnus	21
20	Affichage Liste des CV format XML inconnus	21

1 Introduction

Ce projet est la continuation des TPs effectués à l'université. L'objectif est de finaliser et concevoir un service RESTfull exploitable qui répond au cahier des charges donné. Nous devons mettre en place des APIs REST pour la gestion des CVs :

- Stocker les CV dans une base de données
- Consulter l'ensemble des CV en XML et HTML
- Obtenir des informations sur un CV spécifique en utilisant son ID en XML et HTML
- Être en mesure de supprimer un CV
- Créer une page d'assistance pour les utilisateurs.

À la fin de la mise en œuvre de ce service il est demandé d'effectuer un déploiement sur Clever Cloud.

Vous pouvez accéder au projet en suivant ces liens :

- GitHub : <https://github.com/toubal77/cv24v1>
- Clever Cloud : <https://cv24-toubal.cleverapps.io/>

Liste des acronymes

Terme	Signification
TP	Travaux pratiques
CV	Curriculum vitae
REST	Representational State Transfer
XML	eXtensible Markup Language
HTML	HyperText Markup Language
JDK	Java Development Kit
ID	Identifiant
JAXB	Java Architecture for XML Binding
URL	Uniform Resource Locator
JSP	JavaServer Pages
CSS	Feuilles de style en cascade
JPA	Jakarta Persistence API

2 Configuration requise

2.1 Application déployée

Aucune configuration n'est nécessaire pour accéder au projet déployé. Il suffit d'utiliser l'URL suivante : <https://cv24-toubal.cleverapps.io/>

2.2 Application déployée en local

Avant de commencer, assurez-vous d'avoir les configurations suivantes dans votre machine :

- Environnement : Ubuntu / Windows
- Java : JDK 17
- Spring Boot : 3.2.x
- Maven pour la gestion des dépendances
- MySQL : 8.0.36
- Docker

Après avoir mis en place ces configurations, il vous reste une dernière étape à suivre configurer la base de données en modifiant les paramètres le nom d'utilisateur, le mot de passe et l'URL pour correspondre à vos propres informations, cela dépend de la manière dont vous souhaitez exécuter le projet que ce soit en local ou avec Docker. La figure 1 présente un exemple de configuration

```
## config bdd docker
# spring.datasource.url=jdbc:mysql://db:3306/cv24v1
# spring.datasource.username=root
# spring.datasource.password=root
# spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
## config bdd local
# spring.datasource.url=jdbc:mysql://localhost:3306/cv24v1
# spring.datasource.username=toubal
# spring.datasource.password=toubal
# spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

FIGURE 1 – Configuration de la base de données

Vous avez deux façons de lancer le projet en local : avec la commande Docker 'sudo docker-compose up --build' ou avec 'mvn spring-boot :run'. Par la suite, l'application sera accessible sur 'http://localhost:8080/'.

3 Conception

Le projet est structuré en 7 packages : Model, Controllers, Repositories, Util, Exceptions, Mapper et Services.

- Model : contient les modèles utilisés pour le service REST.
- Util : contient les exceptions, les fichiers de configuration et d'autres fichiers utils.
- Controllers : comprend tous les contrôleurs avec leurs différentes méthodes (GET, POST et DELETE).
- Repositories : contient le Repository pour le CV
- Mapper : regroupe les fichiers de mapping
- Services : comprend les fichiers de service

Le répertoire des ressources contient :

- Les fichiers "xsd" qui contiennent le fichier xsd pour la validation des CV, ainsi que les fichiers "html" pour les pages d'erreur.

Le répertoire "webapp" contient les pages HTML (fichiers JSP et HTML), les fichiers CSS et les images.

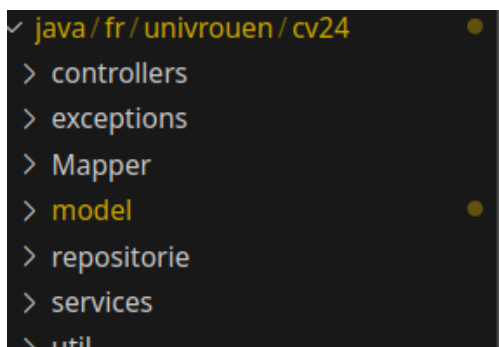


FIGURE 2 – Packages

3.1 Model

Dans le package "Model", il existe deux types de fichiers les fichiers simples avec le nom de l'élément dans le XML et les fichiers de type. Les fichiers simples sont conformes à la CV24 décrite lors du premier TP. Ils permettent de réaliser un mappage de données ("Mapping") du XML vers l'objet JAVA. Les fichiers de type servent à faire le mapping de l'objet vers le XML et permettent également d'aider JAXB à lire le XML et à mapper les tables de la base de données pour chaque élément. Ces fichiers sont orientés dans la gestion de la persistance des données et de la sérialisation/désérialisation XML.

Le fichier package-info permet de fournir documentation et les annotations utilisées dans le package.

3.1.1 Architecture

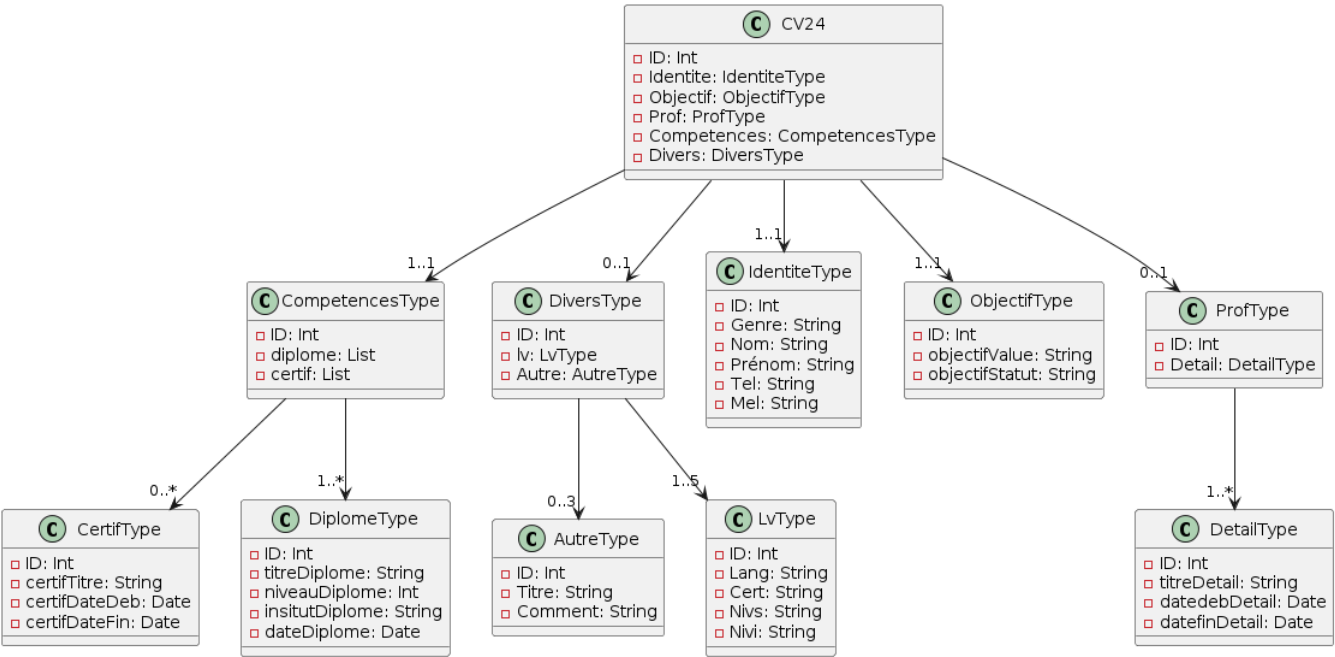


FIGURE 3 – Diagramme de classe

3.1.2 Description des fichiers

Description de l'entité AutreType

Champ	Type	Description
ID	Int	Identifiant
Titre	String	Titre de autre
Comment	String	Commentaire de autre

TABLE 1 – Description de la section Autre

Description de l'entité CertifType

Champ	Type	Description
ID	Int	Identifiant
certifTitre	String	Titre de la certification
certifDateDeb	Date	Date de début de la certification
certifDateFin	Date	Date de fin de la certification

TABLE 2 – Description de la section Certif

Description de l'entité CompetencesType

Champ	Type	Description
ID	Int	Identifiant
diplome	List	Une liste de diplôme
certif	List	Une liste de certifications (certif)

TABLE 3 – Description de la section Competences

Description l'entité CV24

Champ	Type	Description
ID	Int	Identifiant
Identite	Object	Fait référence à la section identite
Objectif	Object	Fait référence à la section objectif
Prof	Object	Fait référence à la section prof
Competences	Object	Fait référence à la section competences
Divers	Object	Fait référence à la section divers

TABLE 4 – Description de la section CV24

Description de l'entité DetailType

Champ	Type	Description
ID	Int	Identifiant
titreDetail	String	Titre du détail
datedebDetail	Date	Date début du détail
datefinDetail	Date	Date fin du détail

TABLE 5 – Description de la section Détail

Description de l'entité DiplomeType

Champ	Type	Description
ID	Int	Identifiant
titreDiplome	String	Titre du diplôme
niveauDiplome	Int	Niveau du diplôme
insitutDiplome	String	Institut ou le diplôme a été obtenu
dateDiplome	Date	Date d'obtention du diplôme

TABLE 6 – Description de la section Diplôme

Description de l'entité DiversType

Champ	Type	Description
ID	Int	Identifiant
lv	Object	Fait référence à la section lv (langues)
Autre	Object	Fait référence à la section autre

TABLE 7 – Description de la section Divers

Description de l'entité **IdentiteType**

Champ	Type	Description
ID	Int	Identifiant
Genre	String	Fait référence genre de l'utilisateur
Nom	String	Nom de l'utilisateur
Prénom	String	Prénom de l'utilisateur
Tel	String	Numéro de téléphone de l'utilisateur
Mel	String	Mail de l'utilisateur

TABLE 8 – Description de la section Identite

Description de l'entité **LvType**

Champ	Type	Description
ID	Int	Identifiant
Lang	String	Langue parlée par l'utilisateur
Cert	String	Certification de la langue
Nivs	String	Niveau de la certification de langue
Nivi	String	Score de la certification de langue

TABLE 9 – Description de la section Lv (langues)

Description de l'entité **ObjectifType**

Champ	Type	Description
ID	Int	Identifiant
objectifValue	String	Intitulé de l'objectif
objectifStatut	String	Statut du poste

TABLE 10 – Description de la section Objectif

Description de l'entité **ProfType**

Champ	Type	Description
ID	Int	Identifiant
Detail	Object	Une liste de details (expériences professionnelles)

TABLE 11 – Description de la section Prof

Description de l'entité **Response**

Champ	Type	Description
ID	Int	Identifiant du document XML
Status	Enum	OK, DUPLICATED, INSERTED, DELETED, UPDATED, ERROR, NONE
Description	String	Message d'erreur ou succès

TABLE 12 – Description de la section Response

3.2 Controllers

Dans le package "Controller", il y a 3 fichiers, chacun correspondant à une méthode HTTP différente : GET, DELETE et POST. Chaque fichier gère les requêtes spécifiques de sa méthode. Par exemple, le fichier GET est utilisé pour récupérer des données, le fichier POST pour envoyer des nouvelles données, le fichier DELETE pour supprimer des données.

Méthode	Nom de la méthode	Paramètre	Description
GET	Index	-	Affiche page d'accueil
GET	Help	-	Affiche page d'aide
GET	Error	-	Affiche pag d'erreur
GET	Resume	-	Affiche tous les CV en format HTML
GET	ResumeXML	-	Affichage tous les CV en format XML
GET	cvXML	Id String	Affiche un CV sous forme XML
GET	cvHTML	Id String	Affiche un CV sous forme HTML

TABLE 13 – Description du controller Get

Méthode	Nom de la méthode	Paramètre	Description
DELETE	delete	Id String	Supprime un CV

TABLE 14 – Description du controller Delete

Méthode	Nom de la méthode	Paramètre	Description
POST	insert	Cv24 CV24	Insertion dans la BDD

TABLE 15 – Description du controller Post

3.3 Repositories

Le package contient un seul fichier. La classe `CVRepository` est une interface qui aide à gérer les CV dans la base de données. En utilisant Spring Data JPA, elle fournit des méthodes pour effectuer des opérations dans la BDD.

3.4 Util

Fichier	Description
CV24ListWrapper	Sert à encapsuler une liste de CV24 pour la sérialisation et la désérialisation XML
DisplayMessageException	Exception personnalisée qui gère les erreurs et permet de charger un modèle HTML d'erreur pour afficher le message
TransformToXML	Contient des méthodes pour transformer des objets CV24 en XML et HTML à l'aide de JAXB et XSLT
Valideur	Contient une méthode pour valider un objet CV24 par rapport à un schéma XSD
WebMvcConfig	Configuration qui définit la résolution de vues pour les fichiers JSP et HTML Thymeleaf

TABLE 16 – Description des fichiers du package 'util'

3.5 Exceptions

Le package contient une seule classe `GlobalExceptionHandler` est un gestionnaire d'exceptions global qui intercepte les exceptions et affiche une vue "error".

3.6 Mapper

Le package contient des classes pour chaque model specifier avant, pour mapper des objets simples vers des objets de type et vice-versa "Mapping".

3.7 Services

Dans le package de services, le fichier `CV24Service` abrite des méthodes responsables de la gestion de nos CV. Ces méthodes sont sollicitées par nos contrôleurs pour exécuter diverses opérations telles que l'insertion dans la base de données, la suppression et l'affichage.

Ce package comprend également la classe `JaxService`, qui offre des fonctionnalités de conversion d'objets Java en format XML en utilisant la technologie JAXB. Elle propose des méthodes pour marshalliser et unmarshalliser, souvent appelées data binding.

4 Déploiement

Le déploiement a été réalisé dans le cloud Clever. Tout d'abord, on a commencé à créer une application de Java + Maven. Le projet se trouve dans un dépôt github. L'hébergement est connecté au code source de la branche Master, l'application récupère le code de la branche Master et le déploie sur le serveur.

En ce qui concerne la base de données, une base de données MySQL a été créée dans le Clue Cloud, reliée à notre application. Vous trouverez dans la figure 4 la configuration nécessaire qui a été faite pour la réalisation du déploiement. Pour obtenir les informations de la base de données, nous avons utilisé les variables d'environnement fournies par l'hébergeur.

```
spring.datasource.url=jdbc:mysql://${MYSQL_ADDON_HOST}:${MYSQL_ADDON_PORT}/${MYSQL_ADDON_DB}
spring.datasource.username=${MYSQL_ADDON_USER}
spring.datasource.password=${MYSQL_ADDON_PASSWORD}
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

FIGURE 4 – Configuration de la base de données

4.1 Requête réalisée

4.1.1 Insertion

Pour ajouter un ou plusieurs CV, on doit utiliser le lien suivant : <https://cv24-toubal.cleverapps.io/cv24/insert> avec la méthode POST. Le corps de la requête doit contenir un flux XML conformément au schéma XSD.



FIGURE 5 – Insertion d'un flux XML

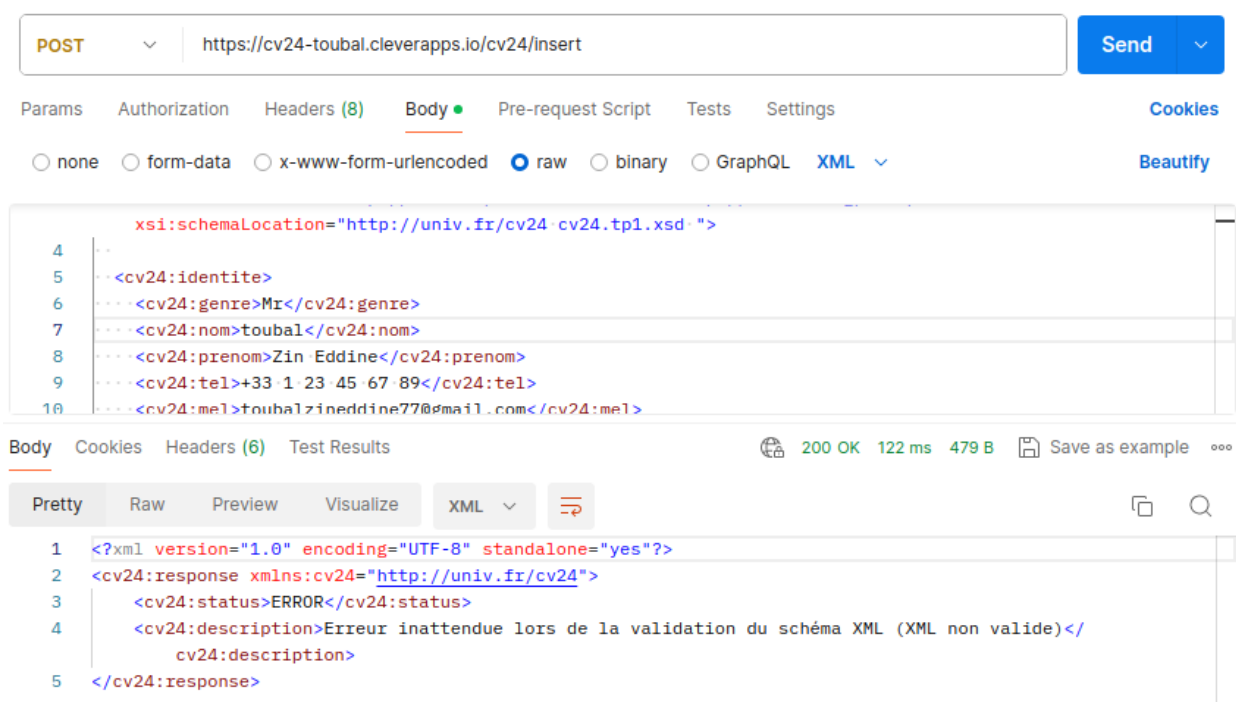


FIGURE 6 – Insertion d'un flux XML non valide

4.1.2 Page help

Pour afficher la page contenant les informations d'aide, on doit utiliser le lien `https://cv24-toubal.cleverapps.io/help` avec la méthode GET

La Page affiche la liste des opérations gérées par notre service REST.

URL	METHODE ATTENDUE	ACTION
/ ou /index	GET	Retourne la page d'accueil du projet
/help	GET	Retourne la liste des opérations gérées par le service REST
/cv24/resume/xml	GET	<p>Retourne une liste résumée les CV24 présentes dans la base en flux XML avec les informations suivantes :</p> <ul style="list-style-type: none">• Identifiant de la CV24• Identité : Genre, Prénom, Nom• Objectif avec le statut spécifié en début de ligne• Diplôme le plus élevé (ou le plus récent) <p>En cas de succès, un flux XML conforme au schéma sera retourné. En cas d'échec, un XML de retour contenant l'identifiant et le statut de l'erreur sera retourné.</p>
/cv24/resume	GET	<p>Retourne une liste résumée les CV24 présentes dans la base en flux HTML avec les informations suivantes :</p> <ul style="list-style-type: none">• Identifiant de la CV24• Identité : Genre, Prénom, Nom• Objectif avec le statut spécifié en début de ligne• Diplôme le plus élevé (ou le plus récent) <p>En cas de succès, le contenu sera retourné au format HTML. En cas d'erreur, un message d'erreur HTML sera affiché</p>
/cv24/xml/{id}	GET	<p>Opération Retourne le contenu complet du CV24 dont l'identifiant est {id}</p> <p>Retour Flux XML conforme au schéma XSD d'une CV24. En cas de succès, un flux XML conforme au schéma sera retourné. En cas d'échec, un XML de retour contenant</p>

Tous droits réservés © 2024

FIGURE 7 – Page d’aide

4.1.3 Page d’accueil

Pour afficher la page d’accueil, on doit utiliser lien : [https ://cv24-toubal.cleverapps.io](https://cv24-toubal.cleverapps.io) avec la méthode GET.

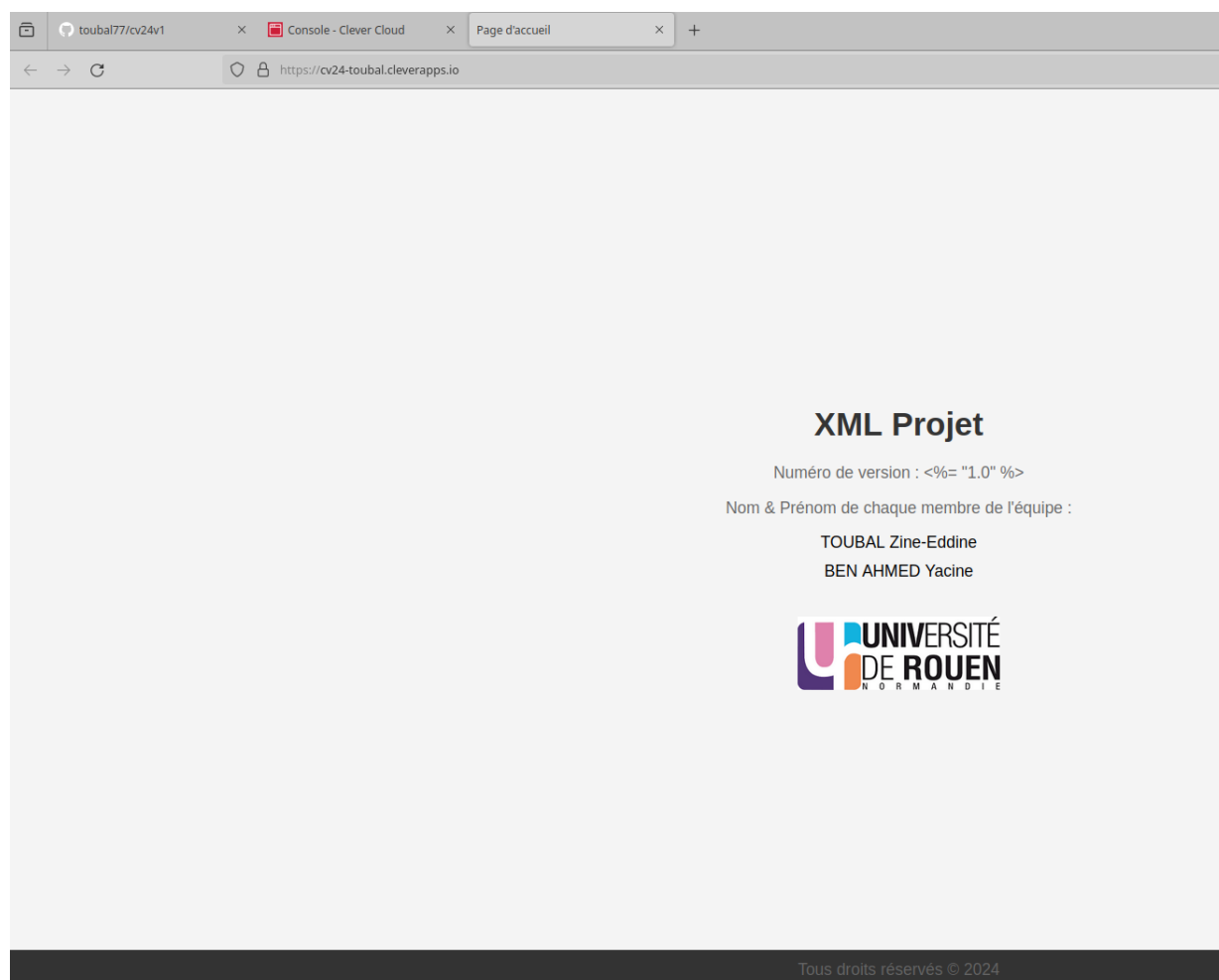


FIGURE 8 – Page d'accueil

4.1.4 Page d'erreur

Pour afficher la page d'erreur, on doit utiliser lien : <https://cv24-toubal.cleverapps.io/help> avec la méthode GET, ou demande n'importe qu'elle page qui n'existe pas dans notre application vous serez redigérer vers cette page

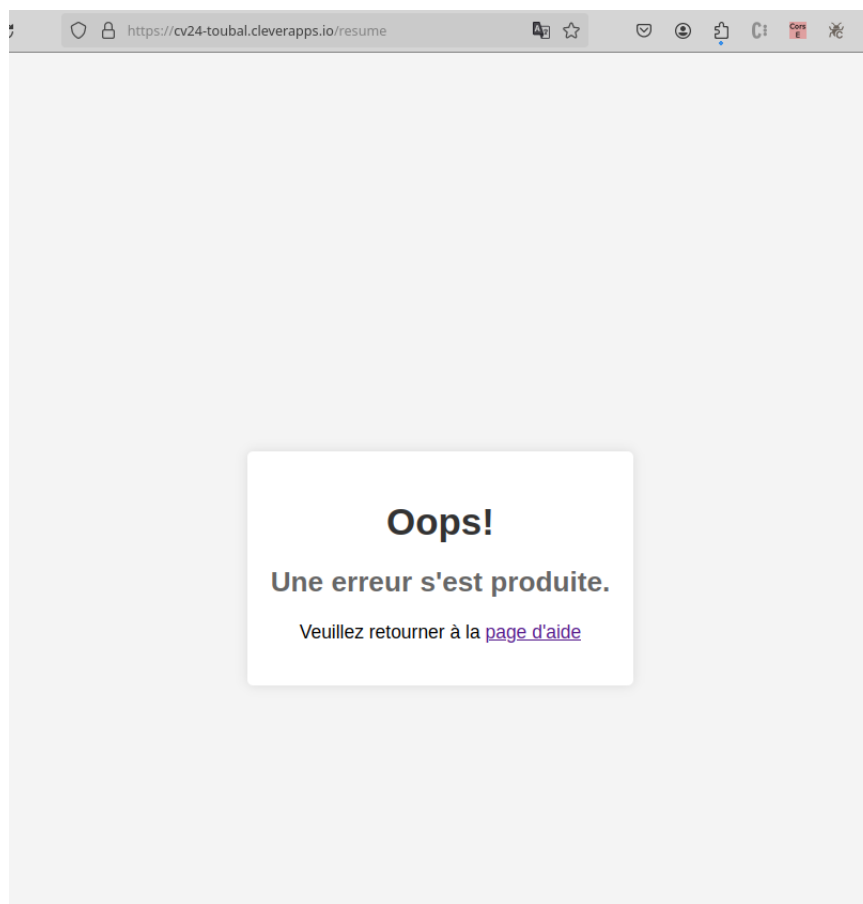


FIGURE 9 – Page d’erreur

4.1.5 Liste des CV format XML

Pour afficher la liste résumée des CVs stockées en format XML valide, on doit utiliser le lien suivant : `https://cv24-toubal.cleverapps.io/cv24/resume/xml` avec la méthode GET

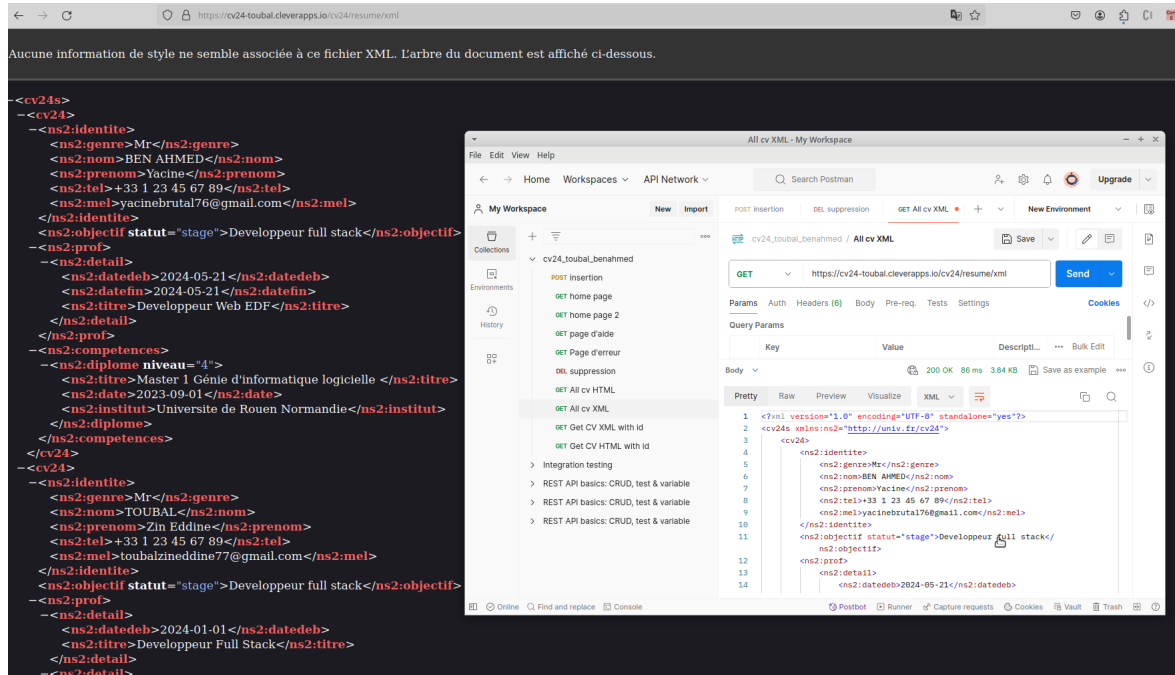


FIGURE 10 – Liste des CV format XML

4.1.6 Liste des CV format HTML

Pour afficher la liste résumée des CVs stockées en format HTML, on doit utiliser le lien suivant : <https://cv24-toubal.cleverapps.io/cv24/resume> avec la méthode GET

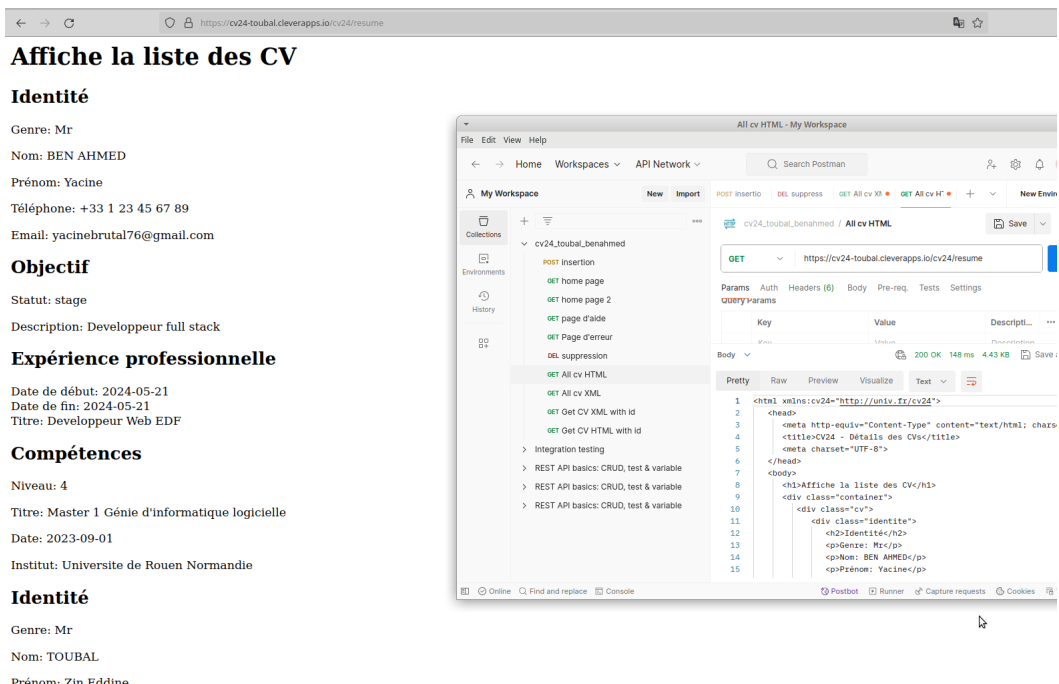


FIGURE 11 – Liste des CV format HTML

4.1.7 CV format XML

Pour avoir le contenu complet d'un CV on spécifiant l'identifiant du CV, on utilise le lien <https://cv24-toubal.cleverapps.io/cv24/xml?id=2> avec la méthode GET. Le retour flux d'un XML conforme au schéma XSD.

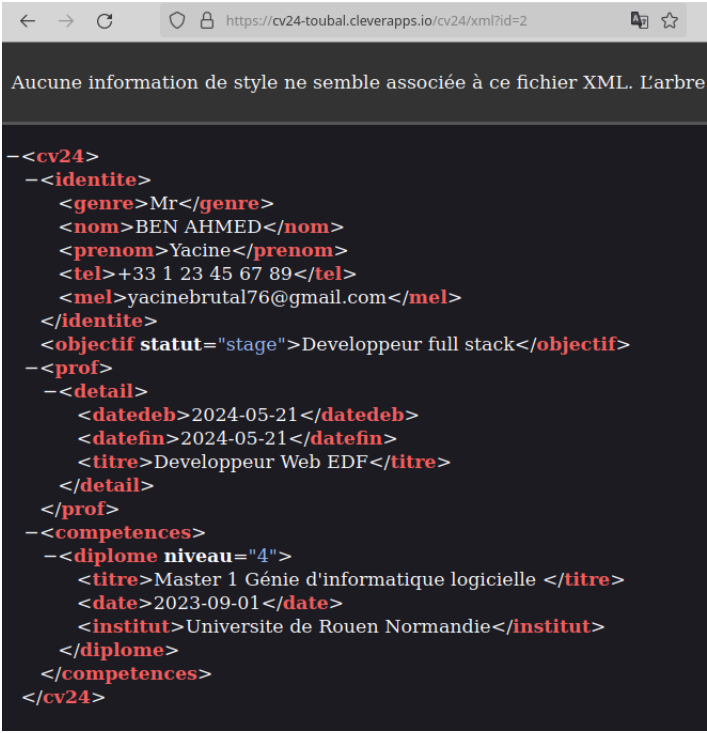


FIGURE 12 – Détail d'un CV format XML

4.1.8 CV format HTML

Pour avoir le contenu complet d'un CV on specifiant l'identifiant du CV, on utilise le lien <https://cv24-toubal.cleverapps.io/cv24/html?id=3> avec la méthode GET. Le retour flux d'un HTML



FIGURE 13 – Détail d'un CV format HTML

4.1.9 Suppression d'un CV

Pour la suppression d'un CV avec identifiant, on utilise le lien suivant : <https://cv24-toubal.cleverapps.io/cv24/delete?id=1> avec la méthode DELETE. le retour est un flux XML.

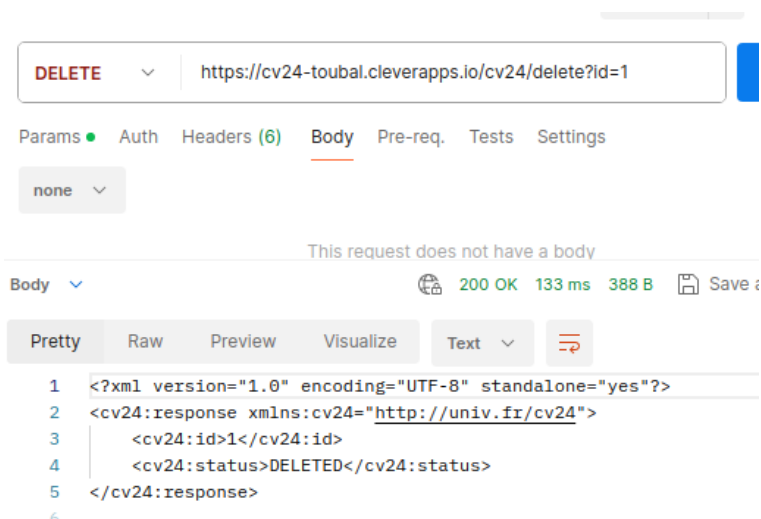


FIGURE 14 – Suppression d'un CV

4.1.10 Message d'erreur

Si un CV n'est pas disponible dans la base de données, bien aucun CV n'est dans la base de données ou d'autres actions qui peuvent provoquer une erreur. une page d'erreur s'affiche qui peut être en XML ou HTML cela dépendra de la requête

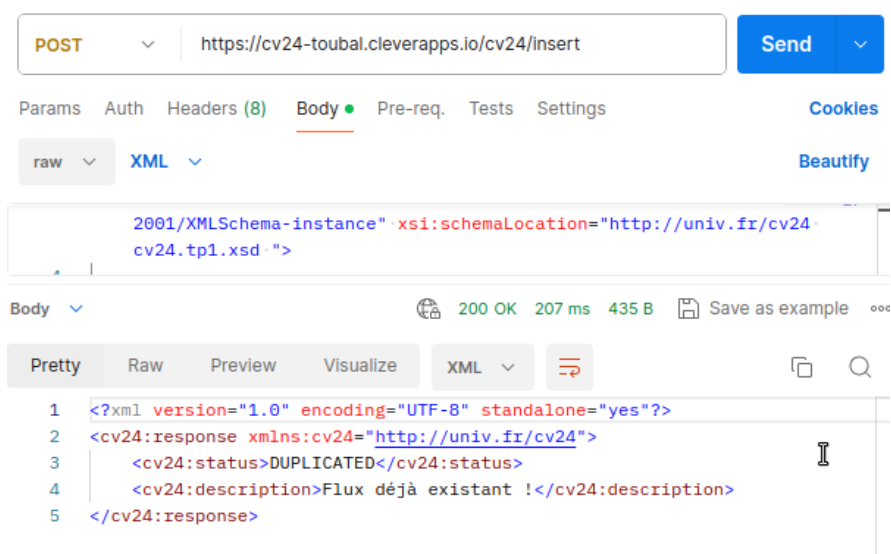


FIGURE 15 – Erreur flux déjà existant

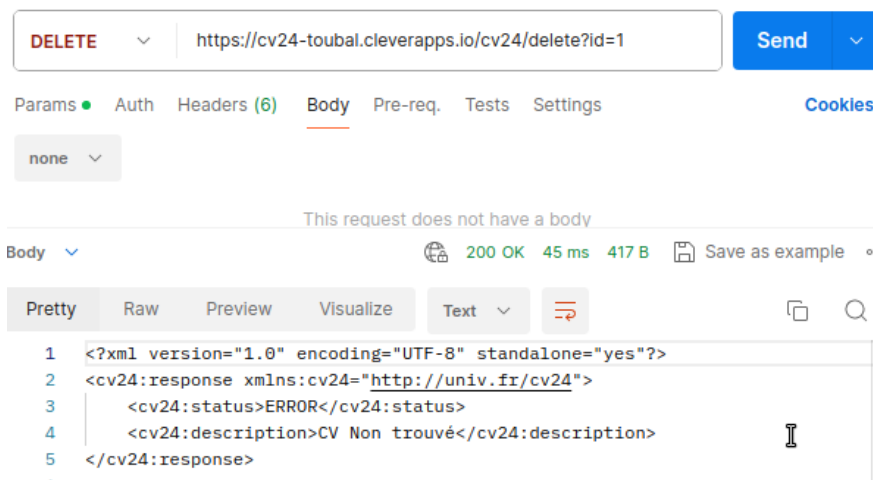


FIGURE 16 – Suppression CV inconnu

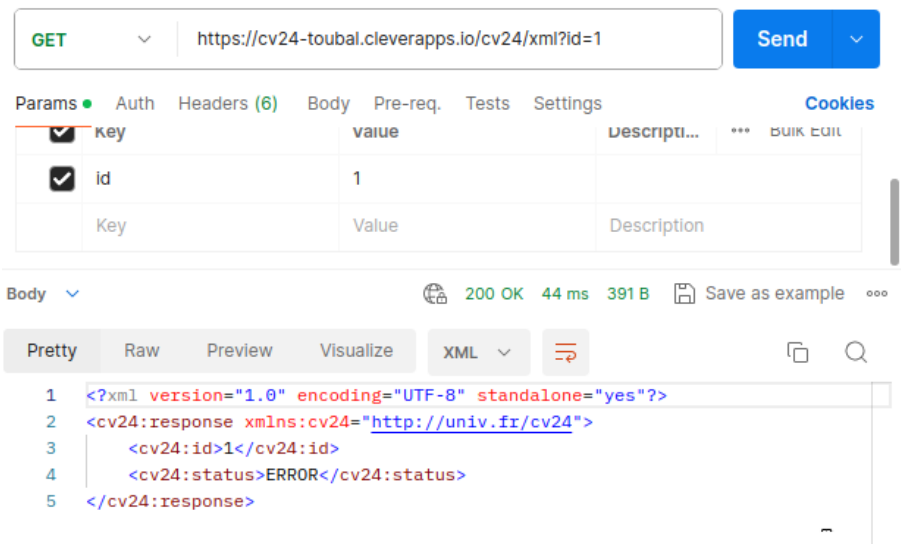


FIGURE 17 – Affichage format XML d'un CV inconnu

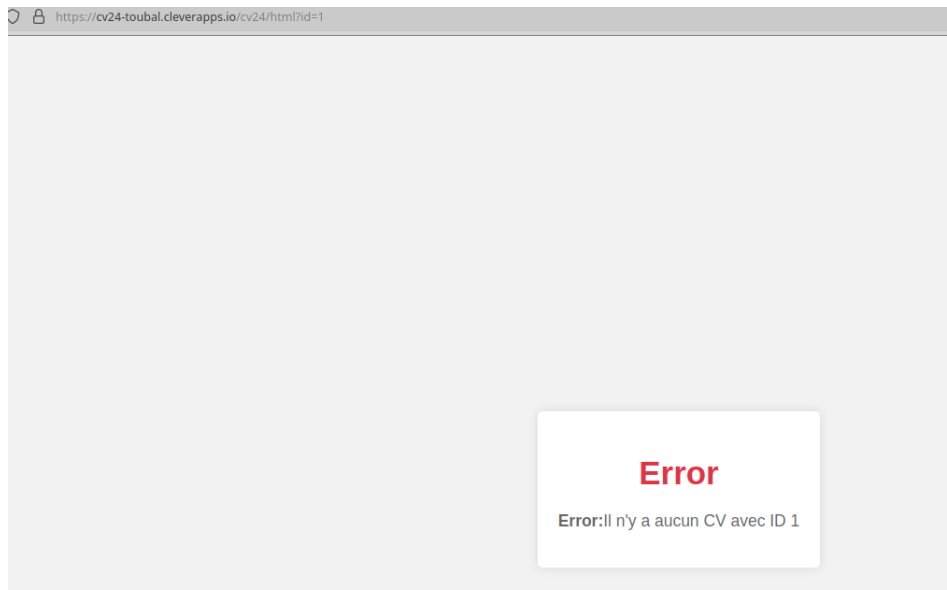


FIGURE 18 – Affichage format HTML d'un CV inconnu

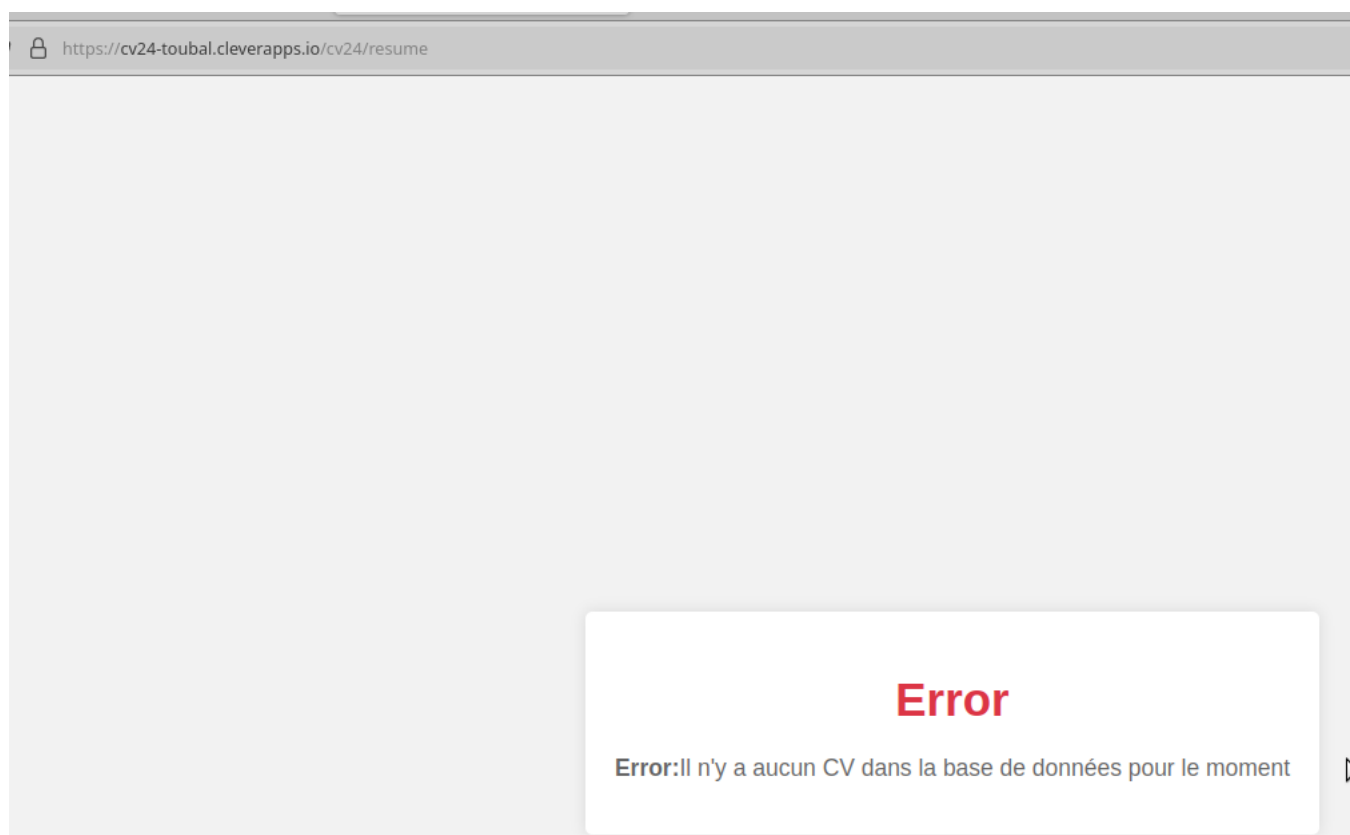


FIGURE 19 – Affichage Liste des CV format HTML inconnus

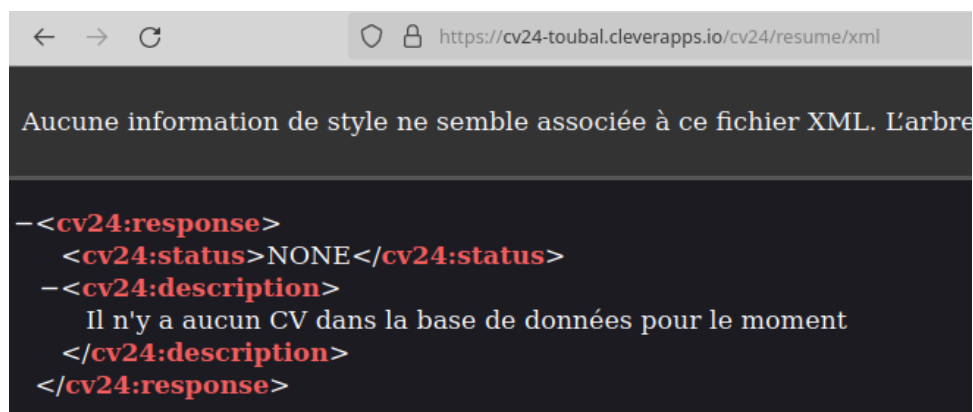


FIGURE 20 – Affichage Liste des CV format XML inconnus

5 Références

1. DZone. (s. d.). Creating a Web Application with Spring Boot. Récupéré sur <https://dzone.com/articles/creating-a-web-application-with-spring-boot>
2. Stack Overflow. (2021). How to Use Both Thymeleaf and JSP in Same Spring Boot Project. Récupéré sur <https://stackoverflow.com/questions/70484274/how-to-use-both-thymeleaf-and-jsp-in-same-spring-boot-project>