

Projet - Service Rest

Cornilleau Titouan
Lassire Eden

Introduction	3
Choix Techniques	4
I - Base de données	4
II - Structure du modèle CV21	4
III - Requêtes	4
1 - Récupération de l'ensemble des CV	4
2 - Créer un nouveau CV	4
3 - Affichage de la page d'accueil	5
4 - Affichage de la page d'aide	5
commande GET : {{url}}/help	5
5 - Affichage d'un cv au format html	5
6 - Affichage d'un cv au format xml	6
7 - Suppression d'un cv	6
8 - Mise à jour d'un cv	6
IV - Difficultés rencontrées	6

Lien git : <https://github.com/cornitit/cv21rest>

Introduction

Afin de clôturer le second semestre dans la matière *langages Web 2*, il nous a été demandé de réaliser un service RESTfull.

Pour ce faire, nous avons utilisé les méthodes vues en cours et en travaux pratiques dans le but de réaliser un gestionnaire de CV. Nous nous sommes appuyés sur nos connaissances en HTML, XML, java, ce que nous avons appris sur l'hébergeur CleverCloud, les différentes méthodes de gestion de bases de données NoSql, SrpingBoot et les frameworks utilisés tels que Log4J, JacksonXML, JSON...

Choix Techniques

I - Base de données

Lors de ce projet, l'un des choix auxquels nous avons été confrontés se trouvait être la façon dont la base de données serait gérée. En effet, il nous fallait un moyen de stocker les différents CV.

C'est dans ce but que nous avons exploré les possibilités décrites dans l'énoncé du projet. Pour ce qui est de JDBC, l'installation était assez laborieuse et nous nous serions retrouvés avec un nombre de tables conséquent. Faire des requêtes sur cette base se serait avéré plus compliqué que ça n'aurait dû être.

Ensuite, il y avait l'option d'utiliser JPA Hibernate. Le point négatif avec cette méthode était le temps que prenait la description de chaque objet. Nous avons donc continué à chercher une solution plus rapide afin d'économiser du temps.

Enfin, le dernier moyen présenté était MongoDB, c'est une base de données NOSQL, c'est-à-dire que les informations stockées sont des objets JSON. Cette base de données nous a paru être la plus simple d'utilisation, car il suffisait de convertir notre cv en JSON afin de le stocker.

Afin de visualiser notre base de données, nous avons eu recours à l'outil Robo3T.

II - Structure du modèle CV21

Pour ce qui est des CV, nous avons repris les exemples vus en cours. Nous sommes partis d'un exemple de CV, et nous avons transformé les différents attributs en objets JAVA. De cette manière, nous avons accès à toutes les informations et pouvons les récupérer sans soucis.

Le projet est basé sur un modèle MVC (modèle, vue, contrôleur). Les CV sont situés dans le modèle. Chaque information est représentée sous forme de String.

III - Requêtes

{{url}} est une variable locale que nous avons définie dans Postman. Elle correspond dans notre cas à : <http://localhost:6039>

1 - Récupération de l'ensemble des CV

commande GET : <{{url}}/resume>

2 - Créer un nouveau CV

commande POST : <{{url}}/insert>

BODY XML (*exemple de body utilisé*)

```
<?xml version="1.0" encoding="UTF-8"?>
<p:cv21 xmlns:p="http://univ.fr/cv21" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
  <p:identite>
    <genre>M</genre>
    <nom>cornilleau</nom>
    <prenom>titouan</prenom>
    <tel>0123456789</tel>
    <mel>mail@mail.fr</mel>
  </p:identite>
  <p:objectif emploi="" stage=""/>
  <p:prof>
    <expe>
      <datedeb>2001-01-01</datedeb>
      <datefin>2001-01-01</datefin>
      <titre>titre3</titre>
    </expe>
  </p:prof>
  <p:competences>
    <diplome niveau="">
      <date>2001-01-01</date>
      <titre>titre</titre>
      <etab>etab</etab>
    </diplome>
    <certif>
      <datedeb>2001-01-01</datedeb>
      <datefin>2001-01-01</datefin>
```

```
<titre>titre</titre>
</certif>
</p:competences>
<p:divers>
  <lv cert="" lang="" nivi="0" nivs=""/>
  <autre comment="" titre=""/>
</p:divers>
</p:cv21>
```

3 - Affichage de la page d'accueil

commande GET : {{url}}/

4 - Affichage de la page d'aide

commande GET : {{url}}/help

5 - Affichage d'un cv au format html

commande GET : {{url}}/htmlcv?id=608a9bab39938617a0ab45f7
608a9bab39938617a0ab45f7 est un exemple d'identifiant

Malheureusement cette fonctionnalité ne fonctionne pas.

6 - Affichage d'un cv au format xml

commande GET : {{url}}/cv?id=608a9e1a39938617a0ab45f9

Affichage du cv ayant pour identifiant : 608a9e1a39938617a0ab45f9

7 - Suppression d'un cv

commande DELETE : {{url}}/delete?id=608b07719b7556072319af97

*Dans l'exemple, nous supprimons le cv ayant pour identifiant :
608b07719b7556072319af97*

8 - Mise à jour d'un cv

commande PUT : {{url}}/update?id=608b0912f4ca002265830841

*Dans l'exemple, nous mettons à jour le cv ayant pour identifiant :
608b0912f4ca002265830841*

