 S. Hérauville	Langage Web 2 - XML	M1 GIL
	Projet : Service REST	Projet 2024
		page 1 / 6

Objectifs

En route pour la dernière phase de ce module !

1. Déployer un service REST permettant de gérer les documents cv24 conformes à la description définie lors du TP n°1.
2. Valider le fonctionnement du service en exploitant une collection de tests postman

Il reste à finaliser le projet (du moins dans le cadre de ce module) afin de concevoir un service RESTful réellement exploitable, et qui réponde au cahier des charges ci-après.

Remarques :

- Ce projet est une mise en application de vos travaux réalisés en TP. De ce fait, chaque fois que cela est possible, réutilisez au mieux les éléments déjà développés avec les formats XSD (Validation de contenu), XSLT (génération de page HTML), XQUERY (Recherche), ...
- Les éléments non spécifiés explicitement dans le sujet sont laissés à votre appréciation.
- Attention : Un code non fonctionnel, ou des codes identiques (plagiat) ne seront pas évalués.
- Le projet peut être réalisé seul ou par groupe de 2 personnes maximum.

I - Description du service REST

Remarque : Le format de retour de chaque requête doit être respecté et homogène sur l'ensemble de l'application. Lorsque plusieurs formats sont proposés, 1 seul doit être implémenté pour la page concernée.

I.1 - Page d'accueil

URL	/
Méthode	GET
Opération	Affiche la page d'accueil
Retour	Format HTML ou XHTML valide
	Affiche la page d'accueil du projet avec les informations suivantes : <ul style="list-style-type: none"> • Nom de votre projet, • Numéro de version, • Nom & Prénom de chaque membre de l'équipe • Logo de l'Université de Rouen


I.2 - Aide

URL	/help
Méthode	GET
Opération	Affiche la page contenant les informations d'aide
Retour	Format HTML ou XHTML valide
	Affiche la liste des opérations gérées par le service REST.
	Pour chacune des opérations proposées par le service, sont affichés : <ul style="list-style-type: none"> • URL, • Méthode attendue, • Résumé de l'opération (format attendu, format de retour, ...)

I.3 - Liste des CV

I.3.1 - Format XML

URL	/cv24/resume/xml
Méthode	GET
Opération	Affiche la liste des CV stockés
Retour	Flux XML
	Liste contenant la synthèse des informations présentes dans la base.
	Pour chaque CV, ne seront affichés que les informations suivantes :

 S. Hérauvill	Langage Web 2 - XML	M1 GIL
	Projet : Service REST	Projet 2024
		page 2 / 6

- Id : Valeur numérique auto-incrémentée lors de l'enregistrement d'un nouveau CV valide,
- Identité : Genre, Prénom, Nom,
- Objectif avec le statut spécifié en début de ligne,
- Diplôme le plus élevé (ou le plus récent).

I.3.2 - Format HTML

URL /cv24/resume
Méthode GET
Opération Affiche la liste des CV stockés
Retour Page HTML (ou XHTML)
Mêmes informations que précédemment mais présentées au format d'une page HTML
Une mise en page sera réalisée pour obtenir un affichage agréable.

I.4 - Détail d'un CV

I.4.1 - Format XML


URL /cv24/xml?id=<id>
avec <id> = valeur numérique de l'identifiant
Méthode GET
Opération Affiche le contenu complet du CV dont l'identifiant est transmis en paramètre.
Retour Flux XML conforme au schéma xsd cv24, comprenant un seul CV
Intégralité de la spécification dont l'identifiant est fourni
Si l'identifiant est incorrect, retour d'un message d'erreur au format XML contenant :
- id → numéro de l'identifiant demandé
- status → ERROR

I.4.2 - Format HTML

URL /cv24/html?id=<id>
avec <id> = valeur numérique de l'identifiant
Méthode GET
Opération Affiche le contenu complet du CV dont l'identifiant est <id>
Retour Page HTML (ou XHTML)
Intégralité du CV dont l'identifiant est fourni par <id>
Si l'identifiant est incorrect, retour d'un message d'erreur au format HTML contenant :
- id → numéro de l'identifiant demandé
- status → ERROR

I.5 - Ajout d'un CV dans la base

URL cv24/insert
Méthode POST
Transmis Flux XML respectant la spécification cv24 à ajouter, conforme au schéma xsd.
Opération Le flux reçu est validé par le schéma XSD de définition cv24
Si le flux est déjà présent, c'est à dire si les informations genre, nom, prenom et tel sont identiques, alors une indication d'erreur est retournée.
Si l'opération est réussie, alors le CV est ajoutée à la base et sa persistance est assurée.
La valeur de <id> est générée automatiquement, par incrémentation de la dernière valeur enregistrée. Cette valeur doit obligatoirement être unique.
Retour Le flux XML retourné contient les informations suivantes :
- id → numéro d'identifiant attribué à la spécification soumise
- status → INSERTED
En cas d'échec de l'opération, les informations de statut seront retournées

 S. Hérauvillle	Langage Web 2 - XML	M1 GIL
	Projet : Service REST	Projet 2024
		page 3 / 6

- status → ERROR
- detail → INVALID | DUPLICATED ⇒ Selon l'origine de l'erreur

I.5.1 - Suppression d'un CV

URL /cv24/delete
Méthode DELETE
Transmis Valeur de l'identifiant <id>
Opération Suppression du CV dont l'identifiant est <id>
Retour Flux XML
Si l'opération à réussi, retour des informations suivantes :
- id → Numéro d'identifiant du CV qui a été supprimé
- status → DELETED
Si l'opération à échoué
- status → ERROR

I.6 - Contraintes du service REST

Tous les échanges avec le service REST doivent utiliser des formats valides :

- Format HTML ou XHTML dans le cas d'affichage,
- Format XML pour les échanges d'informations,
- Les flux reçus par le service devront être validés par le schéma XSD de description cv24.
- En cas d'erreur de traitement, ou de non conformité du flux XML, le processus de traitement sera arrêté, aucune modification de contenu ne sera stockée, et une réponse sera retournée avec des indications d'erreur adéquates.

II - Travail à réaliser


II.1 - Gestion de la persistance des données

Dans les ressources UniversiTice, vous disposez d'une présentation de 3 bases de données : MariaDB, PostgreSQL et MongoDB. Cette liste n'est pas exhaustive, d'autres solutions sont également envisageables. Avant de finaliser votre choix, renseignez vous sur les possibilités disponibles avec votre hébergeur CleverCloud,

Il existe de nombreux tutoriels sur internet concernant la persistance des données avec SPRING. Vous pourrez trouver des informations pertinentes au bout des liens proposés ci-dessous. N'hésitez pas cependant à compléter¹ ces informations avec d'autres ressources.

- SPRING BOOT
<https://devstory.net/11669/spring-boot>
- SPRING & MariaDB ou PostgreSQL ou Oracle ou SQLServer
 - ... et JDBC
<https://devstory.net/11663/spring-boot-spring-jdbc-et-spring-transaction>
 - ... et JPA
<https://devstory.net/11661/spring-boot-jpa-et-spring-transaction>
 - ... et Hibernate
<https://devstory.net/11665/spring-boot-hibernate-et-spring-transaction>
- SPRING & MongoDB
<https://devstory.net/11773/spring-boot-et-mongodb>

¹ Et éventuellement réactualiser

 S. Hérauvillle	Langage Web 2 - XML	M1 GIL
	Projet : Service REST	Projet 2024
		page 4 / 6

Vous développerez dans votre compte rendu une synthèse permettant de justifier vos choix pour la gestion de la persistance, et sa mise en œuvre.

Remarque : Chacune des solutions de cette présentation (non exhaustive) possède ses avantages et inconvénients. Le choix que vous réaliserez n'aura (ni ne devra avoir) aucune incidence sur la "qualité²" de votre projet.

II.2 - Création des contrôleurs

Ce point a été développé dans le second TP sur le service SPRING. Reprenez cette même démarche pour atteindre les objectifs demandés.

Profitez également de cette étape pour faire un peu de ménage dans votre application, réorganiser les contrôleurs, et supprimer les routes inutiles.

Remarque : Lors des TP3 et TP4 de ce module, vous avez développé une feuille de style XSLT afin de transformer votre flux XML en flux HTML.

II.3 - Validation du schéma XSD

Vous avez défini un schéma XSD lors des 2 premiers TP de ce module.

Réutilisez ce schéma pour valider les flux cv24 reçus, afin de ne les insérer dans la base que s'ils sont conformes aux spécifications.

II.4 - Gestion des erreurs

Vous devrez prendre en compte la gestion correcte des erreurs dans votre application afin que le service reste opérationnel même si une requête incorrecte est transmise.

Cette mise en œuvre pourra prendre plusieurs formes :

- Affichage d'un message (succinct) dans le cas d'un retour sous la forme HTML, ou retour au format XML selon la requête effectuée,
- Enregistrement de logs pour identifier l'activité du serveur et faciliter le debugage.

II.5 - Déploiement de la version finalisée

Votre dernier objectif est bien sûr de déployer votre projet sur un hébergeur, CleverCloud en l'occurrence, afin de valider le fonctionnement grandeur réelle, comme les professionnels du domaine !

Une fois cette tâche accomplie, vous pouvez en retirer la satisfaction de votre réussite, et la félicitation du jury pour votre maîtrise du sujet.

Remarques :

- N'oubliez pas de désactiver la fonctionnalité de debug avant le déploiement.
- Votre application déployée doit comprendre au minimum 2 CV conformes aux spécifications.

Rappel : Pensez à arrêter votre application en fin d'utilisation afin de préserver vos crédits, et ne pas avoir de pénalité pour non respect des consignes.


Remarque : L'application devra être démarrée et opérationnelle pendant la phase de correction. vous serez avertis par mail pour démarrer votre application.

II.6 - Validation Postman

Votre service est déployé, vous pouvez le tester en utilisant les différentes formes de requêtes, et pour cela Postman va vous simplifier la vie.

- Effectuez les différentes requêtes avec Postman pour valider le fonctionnement de votre service. Vous avez la possibilité de préparer vos requêtes et de les enregistrer, ce que vous allez faire et de ce fait pouvoir les inclure dans votre compte rendu.
- Réalisez une collection nommée cv24_<nom> ou <nom> est votre nom, dans laquelle seront

² Qualité = Aptitude à répondre aux besoins exprimés

 S. Hérauvillle	Langage Web 2 - XML	M1 GIL
	Projet : Service REST	Projet 2024
		page 5 / 6

enregistrées vos différentes requêtes permettant d'effectuer un test complet de votre service RESTfull
Vous respecterez les points suivants :

- Chaque requête doit comprendre l'intégralité des paramètres nécessaires (endpoint, verbe HTTP, body, header, ...). Vous pouvez également associer un commentaire pour décrire votre requête.
- Utilisez une variable pour définir votre serveur. Ainsi, il est possible de modifier simplement vos requêtes pour passer de votre service local, à votre service CleverCloud.
- L'ensemble de vos requêtes seront regroupées dans une collection, que vous sauvegarderez au format JSON. Ce fichier sera déposé sur votre github.

Vous trouverez les informations nécessaires dans la documentation Postman :

- <https://learning.postman.com/docs/postman/collections/intro-to-collections/>
- <https://learning.postman.com/docs/postman/variables-and-environments/variables/>

<https://www.toolsqa.com/postman/collections-in-postman/>

II.7 - Finalisation de github

A priori le code de votre dépôt github est à jour puisqu'il est utilisé pour le déploiement sur CleverCloud, mais il reste néanmoins les éléments suivants à compléter :


- Dans un répertoire resources, ajoutez les fichiers suivants :
 - Le fichier de requêtes postman (fichier jcv24_<nom>.json) créé précédemment.
 - 2 documents XML conformes à la spécification cv24. La première version sera une version minimaliste ne comprenant que les éléments obligatoires, le second contiendra les éléments optionnels, ainsi que plusieurs éléments chaque fois que cela est possible.
- L'application serveur doit pouvoir être compilée. Les commandes de compilation (via maven) devront être indiquées explicitement dans le compte rendu, sauf pour les commandes usuelles.
- Le script de création de la base de données sera également fourni sur github
- Si vous avez créé un dépôt privé, il est néanmoins possible d'ajouter 2 collaborateurs, tout en conservant le statut privé du dépôt. Assurez vous alors que votre dépôt soit accessible pour les 2 enseignants de ce module :
 - Etienne REITH : etienne.reith@gmail.com
 - Stéphane HÉRAUVILLE stephane.herauvillle@univ-rouen.fr

II.8 - Organisation pratique

- Le projet rendu sera réalisé individuellement, ou en binôme.
- Les documents doivent comprendre le nom du (ou des) auteur(s) sans ambiguïté.
- Le compte rendu doit comporter toutes les indications nécessaires :
 - Description de l'architecture de votre projet : rôle de chaque classe que vous avez développé,
 - Les informations nécessaires pour accéder à votre projet déployé,
 - Les informations nécessaires pour compiler et exécuter localement votre projet à partir des sources,
 - Toute information que vous jugerez pertinente pour la compréhension de votre travail.
- La documentation doit également être accessible sur votre dépôt github.

Date limite de rendu : Le vendredi **24 mai 2024**

Mise en garde vis à vis du plagiat : Pour rappel, la découverte de plagiat de code (total ou partiel) entre projets peut entraîner leur invalidation et par conséquent, l'invalidation globale du module pour l'ensemble des étudiants concernés.

 S. Hérauville	Langage Web 2 - XML	M1 GIL
	Projet : Service REST	Projet 2024
		page 6 / 6

III - Bonus

Les éléments non explicitement demandés pourront faire l'objet d'un bonus, tel que :

III.1 - Gestion des erreurs

- Le travail demandé comporte obligatoirement une gestion minimaliste des erreurs dues à des requêtes incorrectes. Dans cette partie, vous allez optimiser les messages d'erreurs XML retournés par le service, en utilisant un format uniformisé pour les messages d'erreur retournés.
- Spring propose un mécanisme de gestion centralisée des erreurs que vous pouvez mettre en œuvre.

Exemple : lors de la soumission d'un flux cv24 incorrect, retourner la description de la première erreur détectée.

III.2 - API de recherche : Version de base

URL	/cv24/search?...
Méthode	GET
Opération	Recherche la liste des spécifications répondant aux contraintes exprimées dans la requête <ul style="list-style-type: none"> ◦ date : Liste des spécifications dont la date est identique, ou postérieure à celle indiquée dans la requête. ◦ objectif : Liste des CV dont l'objectif spécifié comprend au moins l'un des mots de la requête.
Retour	Flux XML comprenant la liste des articles répondants aux critères de recherche exprimés. Le format sera semblable à celui utilisé pour présenter la liste des articles (§I.3.1). <ul style="list-style-type: none"> - Si l'opération à réussi, mais que la liste est vide, alors seul le statut sera retourné : - status = NONE - Si l'opération à échoué (format de contrainte incorrect, ..), retourner le statut : - status = ERROR

III.3 - Et la sécurité ?

Le module Spring security permet de gérer les mécanismes de sécurité usuels pour une application web. C'est l'occasion d'explorer cette option.

Si vous mettez en œuvre ce module, vous préciserez dans votre compte rendu les éléments activés, et les différences que cela implique pour l'utilisation de votre service REST. Précisez les incidences sur :

- Le code de l'application
- L'utilisation du service
- Les vérifications ou restrictions imposées par les parties utilisées de ce module. Un test Postman peut être utile pour valider le fonctionnement.
- Référence : <https://spring.io/projects/spring-security>