 S. Hérauvill	Langage Web 2 - XML	M1 GIL
	Projet : Service REST & Client	Projet 2023
		page 1 / 6

## Objectifs

En route pour la dernière phase de ce module !

1. Déployer un service REST permettant de gérer les documents STB23 conformes à la description définie lors du TP n°1.
2. Créer un client permettant d'envoyer et/ou de consommer un document STB23 du service REST.

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 ne sera pas évalué.

## 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"> <li>• Nom de votre projet,</li> <li>• Numéro de version,</li> <li>• Nom &amp; Prénom du (ou des) développeur(se)(s)</li> <li>• Logo de l'Université de Rouen</li> </ul>


### 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"> <li>• URL,</li> <li>• Méthode attendue,</li> <li>• Résumé de l'opération (format attendu, format de retour, ...)</li> </ul>

### I.3 - Liste des STB

#### I.3.1 - Format XML

URL	/stb23/resume/xml
Méthode	GET
Opération	Affiche la liste des spécifications stockées
Retour	Flux XML
	Liste résumée des spécifications présentes dans la base.
	Pour chacune, ne seront affichés que les informations suivantes :

 S. Hérauvillle	Langage Web 2 - XML	M1 GIL
	Projet : Service REST & Client	Projet 2023
		page 2 / 6

- Id : Valeur numérique auto-incrémentée lors de l'enregistrement d'une nouvelle STB enregistrée
- Titre
- Description
- Date de validation de la STB
- Nom de l'entité du client

### I.3.2 - Format HTML

URL /stb23/resume  
Méthode GET  
Opération Affiche la liste des spécifications stockées  
Retour Page HTML (ou XHTML)  
Liste des spécifications présentes dans la base.  
Mêmes informations que précédemment mais présentées au format d'une page HTML  
Les informations seront présentées dans un tableau

### I.4 - Détail d'une spécification

#### I.4.1 - Format XML


URL /stb23/xml/{id}  
{id} = valeur numérique de l'identifiant  
Méthode GET  
Opération Affiche le contenu complet de la spécification dont l'identifiant est {id}  
Retour Flux XML conforme au schéma xsd comprenant une seule STB  
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 /stb23/html/{id}  
{id} = valeur numérique de l'identifiant  
Méthode GET  
Opération Affiche le contenu complet de la spécification dont l'identifiant est {id}  
Retour Page HTML (ou XHTML)  
Intégralité de la spécification 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'une STB dans la base

URL stb23/insert  
Méthode POST  
Transmis Flux XML décrivant la spécification stb23 à ajouter, conforme au schéma xsd.  
Opération Le flux reçu est validé par le schéma XSD de définition stb23  
Si le flux est déjà présent, c'est à dire si les informations titre, version et date sont identiques, alors une indication d'erreur est retournée.  
Si l'opération est réussie, alors la STB 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 :

 S. Hérauvillle	Langage Web 2 - XML	M1 GIL
	Projet : Service REST & Client	Projet 2023
		page 3 / 6

- id → numéro d'identifiant attribué à la spécification soumise
- status → INSERTED

En cas d'échec de l'opération, les informations de statut seront retournées

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

### I.5.1 - Suppression d'une spécification

URL /stb23/delete/{id}  
{id} = valeur numérique de l'identifiant

Méthode DELETE

Opération Suppression de la spécification dont l'identifiant est {id}

Retour Flux XML

Si l'opération à réussi, retour des informations suivantes :

- id → Numéro d'identifiant de la spécification qui a été supprimée
- 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 stb23.
- 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 les indications d'erreur adéquates.

## II - Bonus

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

### II.1 - Gestion des erreurs

- Optimisation des messages d'erreurs XML retournés par le service, avec l'ajout d'un champ description.  
*Exemple : lors de la soumission d'un flux stb23 incorrect, retourner la description de la première erreur détectée.*

### II.2 - API de recherche : Version de base

URL /stb23/search?...


Méthode GET

Opération Recherche la liste des spécifications répondant aux contraintes exprimées dans la requête

- date : Liste des spécifications dont la date est identique, ou postérieure à celle indiquée dans la requête.
- titre : Liste des STB dont le titre comprend au moins l'un des mots de la requêtes.

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).

- 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 status :  
- status → ERROR

 S. Hérauvillle	Langage Web 2 - XML	M1 GIL
	Projet : Service REST & Client	Projet 2023
		page 4 / 6

### III - Client du service REST

#### III.1 - Contraintes imposées pour l'application client

- Application autonome exécutée sur une machine locale, sans serveur,
- L'utilisation doit être ergonomique (menu, bouton option, formulaire, ...) et permettre d'utiliser un flux stb23 conforme (avec des parties optionnelles, ou répétées, ....),
- Les paramètres de connexion au service REST seront paramétrables (url, port, ...),
- Les fonctionnalités offertes seront conformes à la description du service REST,
- La transmission d'un article pourra être effectuée selon au moins l'une des deux méthodes suivantes :
  - Saisie directe par l'utilisateur dans l'IHM,
  - Sélection d'un fichier XML stocké sur la machine locale.
- Après chaque échange avec le service REST, les informations retournées devront être affichées,
- Par défaut, aucune validation de flux XML n'est imposée sur le client.

### IV - Travail à réaliser

#### IV.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. D'autres solutions sont également envisageables.

Avant de finaliser votre choix, renseignez vous sur les possibilités disponibles sur 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 compléter<sup>1</sup> ces informations avec d'autres ressources.


- SPRING BOOT  
<https://o7planning.org/fr/11669/spring-boot>
- SPRING & MariaDB ou PostgreSQL ou Oracle ou SQLServer
  - ... et JDBC  
<https://o7planning.org/fr/11663/tutoriel-spring-boot-spring-jdbc-et-spring-transaction>
  - ... et JPA  
<https://o7planning.org/fr/11661/tutoriel-spring-boot-jpa-et-spring-transaction>
  - ... et Hibernate  
<https://o7planning.org/fr/11665/tutoriel-spring-boot-hibernate-et-spring-transaction>
- SPRING & MongoDB  
<https://o7planning.org/fr/11773/tutoriel-spring-boot-et-mongodb>

Vous développerez dans votre compte rendu une synthèse permettant de justifier vos choix pour la gestion de la persistance.

*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 "qualité"<sup>2</sup> de votre projet.*

<sup>1</sup> Et éventuellement réactualiser

<sup>2</sup> Qualité = Aptitude à répondre aux besoins exprimés

 S. Hérauvill	Langage Web 2 - XML	M1 GIL
	Projet : Service REST & Client	Projet 2023
		page 5 / 6

## IV.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.*

## IV.3 - Validation du schéma XSD

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

Réutilisez ce schéma afin de valider les flux STB23 reçus, avant de les insérer dans la base.

## IV.4 - Gestion des erreurs

Vous devrez prendre en compte la gestion correctes des erreurs dans votre application.

Cette mise en œuvre pourra prendre plusieurs formes :

- Affichage d'un message si page HTML ou retour XML selon la requête,
- Enregistrement de logs pour identifier l'activité du serveur.

## IV.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.*
- *Pensez à arrêter votre application en fin d'utilisation afin de préserver vos crédits*
- *Remarque : L'application devra être démarrée et opérationnelle pendant la phase de correction. vous serez avertis par mail.*

## IV.6 - Validation Postman

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

- Effectuez les différentes requêtes avec Postman pour valider le 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 stb23\_<nom> ou <nom> est votre nom, dans laquelle seront enregistrées vos différentes requêtes permettant 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/>

 S. Hérouville	Langage Web 2 - XML	M1 GIL
	Projet : Service REST & Client	Projet 2023
		page 6 / 6

#### IV.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 :6
  - Le fichier de requêtes postman (fichier `jstb23_<nom>.json`) créé précédemment.
  - 2 documents XML conformes à la spécification `stb23`. 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.
- Vous prévoyez également un répertoire distinct pour le code du client.
- Les 2 applications (client et serveur) doivent pouvoir être compilées. Les commandes de compilation (via maven) seront indiquées explicitement dans le compte rendu
- 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](mailto:etienne.reith@gmail.com)
  - Stéphane HÉRAUVILLE [stephane.herauville@univ-rouen.fr](mailto:stephane.herauville@univ-rouen.fr)

#### IV.8 - Organisation pratique

- Le projet rendu devra être réalisé individuellement, ou en binôme.
- Le document doit comprendre le nom du (ou des) auteur(s) sans ambiguïté.
- Le compte rendu doit comporter toutes les indications nécessaire pour :
  - Comprendre l'architecture de votre projet,
  - 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.
- La documentation doit également être accessible sur votre dépôt github.

Date limite de rendu : Le vendredi **26 mai 2023**

***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 conséquence, l'invalidation globale du module pour l'ensemble des étudiants concernés.***