



Projet de synthèse Master ICONE (M2) U.E. « Architectures logicielles orientées services »

A. BOUJU F. BERTRAND

Année universitaire 2014-2015

1 Description générale

L'idée générale est de créer un service permettant de gérer le stockage et la recherche de documents XML. Le service proposera quatre opérations :

- une première opération `depotDocument` permettra le dépôt de documents sur le serveur. Le document sera accepté s'il est valide par rapport au schéma de référence. L'opération prendra deux paramètres : le nom du fichier et son contenu. Elle retournera un entier représentant l'identifiant du document ou une erreur (via une levée d'exception du côté serveur). Le document reçu sera stocké comme fichier sur le serveur.
- Une seconde opération `rechercheDocument` permettra, à partir d'une liste de mots-clés fournis, de retourner les identifiants et les titres des documents comportant ces mots-clés. Un paramètre précisera si les mots-clés sont reliés implicitement par un « et » ou par un « ou ».
- Une troisième opération `retourneDocument` permettra, à partir d'un identifiant fourni, de retourner le document XML correspondant si celui-ci est stocké sur le serveur.
- Une quatrième opération `generePDF` retournera un fichier PDF généré à partir d'une feuille de style XSL-FO et l'identifiant du document. Le nom du fichier PDF retourné (nécessaire pour recréer le fichier sur le client) sera fourni avec le fichier attaché.

1.1 Le format de document

Les documents gérés devront se conformer à un modèle particulier, défini par un schéma XML. Ce schéma devra permettre de vérifier la validité de documents ayant la structure suivante :

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <article>
4   <titre>Un titre</titre>
5   <resume>Ceci est un résumé</resume>
6   <motcles>
7     <!-- au moins un mot-cle -->
8     <motcle>mot1</motcle>
9     <motcle>mot2</motcle>
10  </motcles>
11  <!-- Pour chaque section, l'attribut 'titre' est obligatoire
12       et le contenu de la section doit etre du type xs:string -->
13  <section titre="introduction">Du texte...</section>
14  <section titre="corps">Du texte...</section>
15  <section titre="conclusion">Du texte...</section>
16 </article>
```

Le schéma devra également prendre en compte les contraintes suivantes :

- le vocabulaire défini appartiendra à l'espace de noms `http://www.doc.org/` ;
- tous les éléments fils d'`article` doivent être présents ;
- les autres contraintes sont précisées dans les commentaires insérés dans le document présenté ci-dessus.

Le traitement des documents, notamment l'extraction des mots-clés, pourra s'effectuer via l'API StAX¹ (Streaming API for XML) dont un exemple de code est fourni dans la classe `StAXValidation`.

1.2 La génération de documents PDF

La feuille de style XSLT que vous définirez devra permettre la création d'un fichier XSL-FO, lui-même conduisant, via FOP, à la génération d'un PDF possédant la présentation suivante :

Titre de l'article (gras, 18 pt, centré)

Résumé : Résumé de l'article (11 pt) Résumé de l'article (11 pt) Résumé de l'article (11 pt) Résumé de l'article (11 pt) Résumé de l'article (11 pt) Résumé de l'article (11 pt).

Mots-clés : motclé1, motclé2, motclé3 (11 pt)

1 Introduction

Du texte en 12 pt...

2 Corps

Du texte en 12 pt...

3 Conclusion

Du texte en 12 pt...

1.3 Le service Web

Dans un souci de simplicité (intégration de l'environnement de développement et du serveur d'applications), le service Web et le client seront développés et déployés avec Netbeans sous forme de deux projets séparés.

La persistance des documents XML gérés se fera par le stockage des fichiers XML sur le serveur. Pour réaliser cela, le plus simple est de créer des objets (`File`) en précisant juste le nom du fichier².

Le service devra également conserver l'indexation des documents entre un arrêt et un redémarrage du service. Cela pourra être réalisé en définissant deux méthodes :

- une première méthode précédée de l'annotation `@PostConstruct` qui désérialisera (si elle existe) les structures de données indexant les documents XML. Cette méthode sera appelée automatiquement par le conteneur avant l'activation du service Web (*deploy*).

1. Cette bibliothèque est disponible en standard à partir de la version 1.5 de Java

2. Pour ceux utilisant l'image Debian fournie, le fichier ainsi créé se trouvera alors dans le répertoire `/home/master/glassfish3/glassfish/domains/domain1/config`

- une seconde méthode précédée de l'annotation `@PreDestroy` qui sérialisera les structures d'indexation. Cette méthode sera appelée automatiquement par le conteneur avant l'arrêt du service Web (*undeploy*).

Ces annotations appartiennent à `javax.annotation` et possèdent un certain nombre de contraintes qu'il est nécessaire de respecter (cf. javadoc). En particulier, la classe implémentant le service devra être déclarée comme un composant EJB sans état (*stateless*³) (cf. code ci-dessous) et devra être créée comme un projet *JavaEE* puis choisir *EJB Module*.

```

1  @Stateless
2  @LocalBean
3  @WebService()
4  public class ClasseImplémentantLeService {
5      ...
6      @WebMethod
7      // une opération du service
8      ... methode1(...) { ... }
9
10     @PostConstruct
11     // methode appelée avant le démarrage du service
12     void demarrage() { ... }
13
14     @PreDestroy
15     // methode appelée avant l'arrêt du service
16     void arret() { ... }
17 }

```

Concernant le transfert de fichiers entre le client et le serveur, il sera effectué sous la forme d'un tableau d'octets (`byte[]`).

2 Livrables

Le projet s'effectuera **en binôme** ou, *exceptionnellement*, en monôme. Dans le cas d'un travail en monôme, l'évaluation tiendra compte de cette situation.

Le travail à rendre devra inclure :

- une documentation (.pdf) comprenant une présentation du projet, l'architecture des classes développées, les aspects techniques propres au projet et la manière d'utiliser votre application.
- les deux projets Netbeans (serveur et client) permettant le déploiement du service sous Netbeans qui devront également inclure :
 - le code source **commenté** des classes Java développées⁴ ;
 - dans le projet client, une classe sera créée par opération à tester.
 - le schéma XML et la feuille de style XSLT ;
 - la documentation générée (javadoc) à partir des classes Java ;

Le dossier projet compressé au format **zip**, *nommé avec les deux noms du binôme*, sera déposé sur la plate-forme Moodle dans la section « Devoir » au plus tard **le dimanche 26 octobre 2014, 23h55 dernier délai (attention : la date passée, il ne sera plus possible de déposer des fichiers)**.

D'autre part, lors de la dernière séance de TP consacrée au projet (vendredi après-midi), un point sera fait concernant l'avancement des travaux de développement et l'implémentation des différentes fonctionnalités demandées. Dans ce cadre, pensez à adopter une démarche incrémentale (par exemple, faire fonctionner le service web sans persistance, puis l'ajouter dans un second temps) et à conserver deux versions de votre code : l'une contenant les fonctionnalités déjà opérationnelles, la seconde représentant votre version de travail.

3. Cela signifie que la classe ne peut pas posséder de variables d'instance.

4. Le code développé est propre à chaque binôme. Une trop grande ressemblance entre le travail de plusieurs binômes sera considéré comme un travail commun et noté comme tel.

3 Modalités d'évaluation du projet

Ce travail sera évalué selon les critères suivants :

- Qualité de la documentation fournie⁵. Celle-ci doit décrire l'aspect technique du travail et notamment les choix faits concernant les structures d'indexation mais également la manière dont l'application doit être installée et utilisée.
- Qualité des applications fournies (conformité au cahier des charges, fiabilité, description des tests et des résultats attendus/obtenus).
- Qualité du code (commentaires, nom des identificateurs, indentation...).
- Respect des délais et des spécifications de ce projet avec notamment l'organisation du projet en différentes tâches.

4 Conseils

Le travail demandé peut aisément être découpé en tâches indépendantes dont le résultat pourra être assemblé au fur et à mesure de l'avancement du projet. Par exemple, l'exploitation des documents XML (extraction des mots-clés) peut être menée en parallèle avec le développement du service Web. Dans le cadre d'un travail en binôme, sachez vous organiser pour utiliser de manière efficace votre temps de travail.

Penser également à travailler de manière incrémentale en testant les différentes opérations du service à différents stades d'avancement.

5. Pour cet aspect il est conseillé de se mettre à la place d'une personne devant reprendre votre projet et le comprendre sans devoir (ou pouvoir) vous poser de questions.