

ENSEIRB

Programmation Web / XML

Evaluation 2/2

Version 1.00 du 9 janvier 2013

Etat : Travail

SOPRA GROUP

Historique :

Version	Date	Origine de la mise à jour	Rédigée par	Validée par
1.0	09/01/2012	Création	Mathieu Lombard	

Préambule :

Cette 2^{ème} évaluation est comparable à la 1^{ère} à la différence que seules des nouvelles fonctionnalités seront demandées lors de cette session.

Sommaire :

1. INTRODUCTION	5
1.1. L'application de test (rappel de l'évaluation 1)	5
1.2. L'envoi des résultats	5
1.3. Indépendance des questions	5
2. NOUVELLES FONCTIONNALITÉS	6
2.1. Renvoi des données « About » sous la forme XML	6
2.2. Validation du flux XML About	6
2.3. Renvoi des données d'une tâche sous forme de properties	6

1. INTRODUCTION

1.1. L'application de test (rappel de l'évaluation 1)

Une application (fournie sur la page : <http://uuu.enseirb-matmeca.fr/~slombardy/ens/IF205>) permettra de tester automatiquement la réalisation des élèves.

Hypothèses de l'application de test :

- Les URLs auxquelles réagit cette WebApp sont strictement identiques à ce qui est demandé dans la présente évaluation

Pour utiliser l'application de test, il faut tout d'abord la décompresser (à l'endroit de votre choix, hors de l'arborescence du serveur web si possible) puis lancer, à la racine, la commande suivante :

```
java -jar EnseirbWebXMLEval1.jar <webAppName> <serverPort> <resultDirPath>
```

- <webAppName> correspond au nom de la WebApp, à priori « EnseirbWebXMLWebapp »
- <serverPort> correspond au port du serveur, par exemple « 8080 »
- <resultDirPath> correspond à un chemin vers un répertoire dans lequel un fichier résultat sera écrit

Il est à noter que l'URL utilisée pour accéder au serveur sera « localhost ».

Par ailleurs, les noms des élèves d'un binôme sont récupérés automatiquement via l'URL /about, il faut donc bien s'assurer que celle-ci est opérationnelle (sinon, le fichier de résultat ne permettra pas d'identifier les élèves et cela sera considéré comme une absence de fichier).

Note importante : dans le cadre de cette évaluation, un test automatique par question (identifiable par le nom de la classe de test dans les traces de l'application) échouera afin de limiter les possibilités de tricherie.

Ainsi, à la fin de l'évaluation, il devra y avoir autant de tests échoués qu'il y a de questions (soit 3 en tout).

1.2. L'envoi des résultats

Une fois les exercices terminés, le fichier comportant les derniers résultats devra suivre 2 étapes :

- Être zippé
- Puis être envoyé à l'adresse suivante : mathieu.lombard@sopragroup.com

Si aucun fichier n'est reçu pour un binôme donné à l'issue du temps de l'évaluation, la note sera alors de 0.

1.3. Indépendance des questions

Les questions de cette évaluation sont indépendantes et peuvent être traitées dans n'importe quel ordre.

2. NOUVELLES FONCTIONNALITÉS

2.1. Renvoi des données « About » sous la forme XML

Dans cet exercice, il s'agit de renvoyer les informations accessibles via la servlet *AboutServlet* au format XML.

Le format des données à renvoyer doit être le suivant :

```
<about studentsNumber='123' group='Gx' class='I2' teacher='t'>
  <student fisrtName='f1' lastName='l1' />
  <student fisrtName='f2' lastName='l2' />
  ...
</about>
```

Les valeurs des attributs doivent correspondre à ce que renvoie les URL */about/action=<action>*. Le nom du *teacher* doit aussi être dynamique en prenant en compte les éventuels changements via */about/teacher/post*.

Il faut alors :

- Compléter le fichier *web.xml* pour renvoyer le flux XML ci-dessous via l'URL */about/xml*
- Compléter la servlet *AboutServlet* pour qu'elle traite la demande

2.2. Validation du flux XML About

Il s'agit ici de créer un schéma XSD pour vérifier la validité du flux ayant la forme de la question précédente. Ce ne sera pas le flux créé dans la question précédente qui sera testé, mais des flux envoyés par l'application de tests.

Il faut alors :

- Compléter le fichier *web.xml* et la servlet *AboutServlet* pour réagir à l'URL */about/xml/isValid*
 - Le flux XML sera envoyé en POST sur cette URL
- Créer un schéma XSD qui validera que
 - tous les attributs sont obligatoires sauf *teacher*
 - il peut y avoir autant de fils *student* que possible
 - l'attribut *studentsNumber* est de type entier et les autres sont tous de type *string*
- Formater puis renvoyer le résultats de la validation via la méthode *XMLToolkit.createPostResult(message, success)*
 - message est une chaîne de caractères indiquant un message libre
 - success est le booléen renvoyé par *XMLToolkit.isXMLValid*

2.3. Renvoi des données d'une tâche sous forme de properties

L'objet de cette question est de formater le flux XML d'une tâche sous la forme d'un fichier .properties en utilisant XSL.

Le format du fichier à renvoyer devra avoir la forme suivante :

```
# <exportFormat>
id=1
title=titre de la tâche
asker=auteur
owner=responsable
description=description de la tâche
status=done / todo
```

Il faut alors :

- Compléter le fichier *web.xml* et la servlet *CRUDTaskServlet* pour réagir à l'URL */task/view/xml?id=<taskID>&exportEncoder=<exportFormat>*
 - Le paramètre *<taskID>* indique l'identifiant de la tâche dont on veut avoir les propriétés
 - Le paramètre *<exportFormat>* indique le texte qu'il faut substituer en haut du flux des propriétés
- Créer une feuille XSL qui formatera le flux XML renvoyé par *XMLMediator.getTask(<taskID>)* selon la description ci-dessus
 - Le paramètre *<exportFormat>* devra être passé en paramètre de *XMLToolkit.transformXML* pour être inclus dans le flux des propriétés
 - Selon la valeur de l'attribut *done* de la tâche, la valeur du status (dernière ligne du flux) devra valoir « todo » (si *done=false*) et « done » (si *done=true*)

Note :

Afin que le moteur XSL n'insère pas de balise XML dans le flux résultat, la feuille XSL devra contenir l'instruction suivante après la balise *xsl:stylesheet* :

```
<xsl:output method="text" omit-xml-declaration="yes"/>
```

De même, la Servlet (qui traite la demande du client) devra indiquer au client (le navigateur) que le flux de données correspond à du texte ; cela s'effectue via l'instruction suivante :

```
response.setHeader("Content-Type", "text/plain");
```

Afin de pouvoir créer un retour à la ligne (nécessaire en fin de chaque ligne), l'instruction suivante pourra être utilisée :

```
<xsl:text>
</xsl:text>
```