Web & XML

ENSEIRB

Programmation Web / XML

Evaluation 2/2

Version 1.00 du 8 janvier 2015

Etat : Travail

SOPRA GROUP

Historique :

Version	Date	Origine de la mise à jour	Rédigée par	Validée par
1.0	08/01/2015	Création	Mathieu Lombard	

***Sommaire* :**

1. INTRODUCTION

1.1. L'application de test (rappel)

Une application (fournie sur la page moodle du cours) 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 a été demandé lors des différents TDs

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 EnseirbWebXMLEval2.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 (qui doit exister) dans lequel un fichier résultat sera écrit

Il est à noter que le nom du serveur utilisé 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 sur les 18 tests) ; si vous avez **15 tests qui passent**, cela signifie donc que vous avez **100% de bonnes réponses**.

1.2. L'envoi des résultats

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

- Être zippé (**sans le renommer** au préalable)
- Puis être envoyé à l'adresse suivante : mathieu.lombard@soprasteria.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. Récupération de données brutes sur les tâches via XPath

Classe de test	RawDataTaskTestCase
Nombre de tests	9

Il s'agit ici de créer une nouvelle *Servlet RawDataTasksServlet* qui répondra aux modes de fonctionnement décrits ci-dessous.

Le travail demandé est de rédiger des requêtes XPath répondant aux cas ci-dessous en les appliquant au flux des Tâches. Il faut ensuite renvoyer les valeurs fournies en utilisant le code suivant (ou la variable <xPath> correspond à l'une des requête demandée) :

```
String tasks = XMLMediator.getTasks();
List<String> filteredTasks = XMLToolkit.getXPathValues(tasks, <xPath>);
String sResponse = StringFormatUtil.sortAndFormatAsCSV(filteredTasks);
response.setHeader("Content-Type", "text/plain");
ServletToolkit.writeResponse(response, sResponse);
```

Pour connaître le fonctionnement demandé, le paramètre action sera envoyé au serveur via l'URL ; sa valeur pourra être récupérée comme cela a été fait pour la Servlet *AboutServlet*.

Avant de pouvoir commencer, il faut référencer cette Servlet dans le fichier *web.xml* pour qu'elle réponde à l'URL : */task/info*.

2.1.1. Récupération des ids des tâches pour un « asker » donné

Une liste d'identifiants de tâches, dont le fils « asker » est spécifié dans l'URL, devra être renvoyée quand l'URL suivante sera invoquée :

- */task/info?action=id4Asker&askerValue=<asker>* ; où asker est le nom de l'émetteur de la tâche

2.1.2. Récupération des ids des tâches pour une priorité donnée

Une liste d'identifiants de tâches, dont l'attribut « priority » est spécifié dans l'URL, devra être renvoyée quand l'URL suivante sera invoquée :

- */task/info?action=id4Priority&prioValue=<prio>* ; où prio est la priorité des tâches dont on veut obtenir les ids.

2.1.3. Récupération du titre d'une tâche à partir de son id

Un champ texte correspondant au titre d'une tâche, dont l'attribut « id » est spécifié dans l'URL, devra être renvoyée quand l'URL suivante sera invoquée :

- */task/info?action=title4Id&idValue=<id>* ; où id est l'identifiant de la tâche

2.2. Formatage des données en CSV

2.2.1. Paramétrage (Rappel de la 1ère évaluation)

Il s'agit ici d'offrir au client la possibilité d'obtenir des données formatées en CSV.

La méthode utilisée sera de passer par une transformation XSL qui, à partir d'un flux XML, renverra du « texte ».

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>
```

Pour information, les espaces et les tabulations sont ignorés lors des comparaisons avec le résultat attendu lors du passage des tests automatiques.

2.2.2. Formatage des statistiques en CSV

Classe de test	StatsCSVTestCase
Nombre de tests	2

Il s'agit ici de permettre à la WebApplication de renvoyer les statistiques selon le format CSV.

Il faut alors:

- Compléter Le fichier web.xml pour que l'URL */stats/csv* soit traitée par la Servelt *StatsServlet*
- Compléter la Servlet *StatsServlet* pour qu'elle gère ce nouveau cas et qu'elle invoque une transformation XSL
- Créer un nouveau fichier XSL qui, à partir du flux des statistiques (obtenu via la méthode *buildStats()* de *XMLMediator*) formate les données comme suit :

```
user;total;done;intime;late
user1;2;0;2;0
user2;2;1;1;0
user3;1;0;0;1
```

  - Où la 1ère ligne correspond aux intitulés des colonnes (texte en dur)
  - Où chaque ligne indique les informations suivantes (dans l'ordre et en séparant chaque donnée d'un point-virgule)
    - Le nom de l'utilisateur
    - Le nombre de tâches qu'il a (ou a eu) à faire (cela correspond à la somme des 3 chiffres suivants)
    - Le nombre de tâches qu'il a déjà fait
    - Le nombre de tâches qu'il n'a pas encore fait et pour lesquelles il n'est pas en retard

- Le nombre de tâches qu'il n'a pas encore fait et pour lesquelles il est en retard
- La liste doit être triée alphabétiquement selon le nom des utilisateurs

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

Classe de test	AboutXMLTestCase
Nombre de tests	7

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 students_number='123' group='Gx' class='I2' teacher='t'>
      <student first_name='f1' last_name='l1' />
      <student first_name='f2' last_name='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