UNIVERSITÉ DE ROUEN

Langages Web 1 – TP 11 2016–2017

Programmation coté serveur avec Java

Introduction

- Une référence de TAG-LIB (la bibliothèque de balises JSP) est disponible à cette adresse : https ://jstl.java.net/
- Le manuel de référence de MySQL est disponible à cette adresse : http ://dev.mysql.com/doc/refman/5.7/en/index.html
- Le manuel de référence de JDBC est disponible à cette adresse : http ://docs.oracle.com/javase/7/docs/technotes/guides/jdbc/

Exercice 1 – *Hello Tomcat !*

Question 1.1 : Qu'est-ce que JSP ? Quel lien y a-t-il avec les servlets ?
Question 1.2 : Écrivez un programme `form.jsp` qui génère deux formulaires en auto-référence (qui envoient les données au même programme) en reprenant le principe utilisé dans le programme `form.php` qui vous a été fourni au TP 06.
Question 1.3 : Ajoutez du code qui affiche :
- le nom de la méthode HTTP utilisée pour l'invocation de la JSP,
- la liste des paramètres reçus à travers la requête HTTP.

Exercice 2 – *Les sessions avec JSP*

Question 2.1 : Que faut-il faire pour lier un programme JSP à la session du client ?
Question 2.2 : Modifiez le programme de l'exercice précédent pour que après chaque requête soient affichées la nouvelle valeur du champs `nom` ainsi que son ancienne valeur.
Question 2.3 : Optionnel : ajoutez un bouton pour détruire la session.

Exercice 3 – *MVC avec JSP*

Pour cet exercice, nous allons également reproduire le travail réalisé en PHP à l'exercice 3 du TP 10.
Question 3.1 : Écrivez les différentes classes correspondant au modèle (`Person`, et `PersonModel` générant une liste de personnes aléatoirement, *cf* TP 10). Il s'agira ici de classes Java simples.
Question 3.2 : Écrivez une JSP `PersonView` qui produira une page contenant tableau HTML contenant les données d'une liste de personnes qui aura été placé dans un attribut de `request` nommé `"pl"`. Une personne sera affichée par ligne) avec les en-têtes `Id`, `Titre`, `Nom` et `Prénom`.
Question 3.3 : Écrivez une servlet qui remplira le rôle de contrôleur, c'est à dire qui recevra et traitera la requête du client. Cette servlet fera appel à votre modèle pour produire une liste de personne et fera appel à votre vue pour afficher cette dernière.
Question 3.4 : Optionnel : reprendre l'architecture de vue proposée au TP précédent :
- une sous-vue (une JSP) sera dédiée à la production du code du tableau HTML uniquement,
- une vue application (une autre JSP) produire le code principale de la page en utilisant la première JSP pour procéder à l'affichage du tableau.

Exercice 4 – *Java et les bases de données*

Nous allons ré-exploiter la base de données du TP précédent pour alimenter notre modèle. La documentation de référence de JDBC est disponible à cette adresse :
docs.oracle.com/javase/6/docs/api/java/sql/package-summary.html
Voici un exemple de connexion à une base de donnée en utilisant JDBC :

```
import java.sql.*;
...

// Chargement du connecteur JDBC
Class.forName("com.mysql.jdbc.Driver");
Connection con = null;
ResultSet rst = null;
Statement stmt = null;
try {
  String serveur = "mysql.exemple.com";
  String login = "toto";
  String motDePasse = "123456";
  String schema = "mabase";
  String url = "jdbc:mysql://␣"+serv+"/"+schema+"?user␣=␣"+login+"&password
     ␣=␣"+ motDePasse;
  con = DriverManager.getConnection(url); // Connexion a la base de donnees
  stmt = con.createStatement(); // Pour pouvoir executer des requetes sur
     la base
}
catch(Exceptione){
  System.out.println(e.getMessage());
}
try {
  // Execution de la requete :
  rst = stmt.executeQuery("select␣*␣from␣etudiants␣where␣nom␣=␣'Dupont'");
  // Recuperation des resultats :
  while(rst.next()){
    String prenom = rst.getString("prenom");  // recuperation d'un attribut
        SQL
    System.out.println(prenom);
  }
}
catch (Exceptione){
  System.out.println(e.getMessage());
}
finally{
  // Fermeture de la connexion :
  rst.close();
  stmt.close();
  con.close();
}
```

Question 4.1 : Créez un modèle `PersonDB.java` dont la méthode `getPersons()` procédera à l'extraction des informations depuis la base de donnés plutôt que de les générer aléatoirement.

Exercice 5 – *Bonus : génération et affichage de contenu sur plusieurs pages*

Question 5.1 : Modifiez votre contrôleur pour qu'il accepte une variable de formulaire indiquant le numéro de la page à afficher.
Question 5.2 : Modifier votre vue pour y ajouter des boutons/liens de navigation (page suivante, page précédente, première page, dernière page) et afficher le numéro de la page en cours.

Question 5.3 : Ajoutez en face de chaque ligne une icône qui permet de supprimer l'entrée. Ajouter le contrôleur effectuant cette action.