NOM FABRE

Prénom Adrien

Promo 2016

Date 9/1/2015

FABRE Adrien
M1 - 2014

14,25

MATIÈRE Java EE

1) briques du Java EE

JSP : Les fichiers JSP correspondent aux vues du modèle MVC.
Ils sont écrits dans le langage JSP qui est proche du HTML mais qui implémente d'autres fonctionnalités comme par exemple le "expression language" pour permettre d'effectuer de la logique.
Ils sont traduits pour renvoyer au client des pages HTML

Servlet : Les servlet correspondent aux contrôleurs du modèle MVC.
Ce sont des classes java, pour traiter des requêtes Http elles héritent de la classe HttpServlet.
Leur rôle est d'"aiguiller" les requêtes faites par le client, d'appeler le code métier effectuant du traitement sur les paramètres des requêtes et de rediriger vers la bonne jsp. La bonne pratique du MVC veut que les JSP ne soient pas accessible autrement que par l'intermédiaire d'une servlet.

EJB : Enterprise JavaBeans.
Ce sont des classes Java qui implémentent de la logique métier comme par exemple la gestion de la persistance en base de données.

2) Technologies de JSP.

Les composantes JSTL (Java Standard Tag Library) et EL (Expression Language) sont des technologies permettant de manipuler de la logique et des objets Java dans les vues JSP. La bonne pratique veut que l'on n'écrive pas de code Java

dans une JSP (séparation vue - modèle/controleur). Le langage EL comporte des balises permettant d'écrire du code Java alors que le JSTL comporte des balises remplaçant ce code Ja

3) Types de portée

- page : la variable n'est accessible que sur une page
- requête : la variable vit tant que la requête est traitée, elle peut par exemple être accédée par deux JSP différentes
- session : la variable vit tant que l'utilisateur n'a pas fermé son navigateur (ou que son inactivité n'a pas été trop longue)
- application : la variable a la même durée de vie que l'execution du programme. Deux utilisateurs connectés à des moments différents peuvent avoir accès à la même variable si l'application n'a pas été fermée entre temps.

4) Java Bean

Un JavaBean est une classe Java destinée au transport de données, il ne doit pas y avoir de code métier dedans. Pour être un bean la classe doit suivre les critères suivants:

- classe publique
- attributs privés liés à un couple accesseur/mutateur publiques
- constructeur par défaut sans argument
- sérialisable : possibilité de persistance

Un exemple de JavaBean:

```
public class MyBean implements Serializable
{
    private int propriété1;
    private String propriété2;

    // si l'on en écrit pas, Java génère un constructeur par défaut sans arguments

    public int getPropriete1 () { return propriété1; }
    public string getPropriete2 () { return propriété2; }
    public void setPropriété1 (int x)    { propriete1 = x; }
    public void setPropriete2 (String x)    { propriete2 = x; }
}
```

5) Bean dans un JSP

Pour récupérer la valeur de nom d'un bean déjà déclaré dans une JSP, deux méthodes possibles sont :

- via l'EL : ${myBean.nom}
- via JSP les scriptlets : <%= myBean.getNom(); %>

1,5

6) formulaire HTML

Pour traiter notre requete, on peut utiliser la méthode proccessRequest qui interceptera les requêtes GET et POST puis traiter les paramètres de la requête contenant les informations :

```
public void ProcessRequest (HttpServletRequest request, HttpServletResponse response)
{
    string couleurFond = request.getParameter ("fond");
    string instructionFond = request.getParameter ("fond2");
    string couleurTexte = request.getParameter ("texte");
    string instructionTexte = request.getParameter ("texte2");
}
```

2

7) création de Servlet : généralement on fait hériter nos servlet de la classe HttpServlet car bien qu'une servlet soit capable de traiter différents types de requêtes, c'est le plus souvent des requêtes Http que nous utilisons.

1

8) paramètres de GetO et de PostO

Ces deux méthodes (que nous avons remplacées par ProcessRequest dans la question précédente) reçoivent deux paramètres qui permettent de recevoir la requête Http du client avec ses paramètres et de forger une réponse. Ces deux paramètres sont de type HttpServletRequest et HttpServletResponse. ✓ (1)

9) Types d'EJB

Les deux seuls types d'EJB sont stateless et statefull.

10) Classe Entity

Les class Entities sont des objets Java destinés à contenir les informations d'un tuple en base de donnée et à gérer la persistence de ces information en ~~faisant~~ permettant des écritures et des lectures en base. (1)
Les Class Entities respectent les propriétés des JavaBean (cf. question 4). On peut donc dire qu'un Class Entity est un JavaBean mais la réciproque n'est pas nécessairement vraie.

12) Persistence - main

Dans la Class Entity de la question 11, la ligne 9 précise qu'en base de donnée ~~la colonne~~ le champs isbn ne peut pas contenir de valeur NULL
Dans ce main, on crée un objet livre via le constructeur par défaut (qui met donc les différent champs à NULL), on initialise les champs titre, prix et description et l'on essaye de l'écrire dans la base de donnée ce qui n'est pas possible puisque le champs isbn est toujours NULL. Il faudrait donc ajouter monLivre.setISBN ("1234567890ABCD"); entre les lignes 6 et 7 ✓ (2)

11) Persistence - Entity

Du fait de la presence de cette question et du nombre de points qu'elle rapporte j'imagine que le code comporte une erreur cependant je n'arrive pas à la detecter et serais donc tenté de répondre qu'il n'y en a pas. ✗