Examen 2ème session - LI 328

Ludovic Denoyer - Sylvain Lamprier - Tommaso Bianco

12 juin 2013

Les documents sont autorisés.

Contexte général

Vous devez développer un site web pour agréger des critiques d'albums de musique.
Des utilisateurs anonymes (non inscrits au site) pourront :
- chercher des albums par nom
- lire les critiques associées aux albums.

Les utilisateurs autorisés (après connexion au site) pourront en plus :
- insérer un nouveau album
- insérer une nouvelle critique d'un album existant.

Les ressources suivantes sont utilisées pour organiser l'information manipulée par le site :
- `user` :
  - `id` : un identifiant unique, alphanumérique
  - `nom` : le nom de l'utilisateur
  - `password` : le mot de passe de l'utilisateur pour se connecter au site
- `album` :
  - id : identifiant unique, numérique ou alphanumérique
  - nom : le nom de l'album
  - groupe : le nom du groupe
- `critique` :
  - `id` : identifiant unique, numérique ou alphanumérique
  - `album` : album concerné par la critique
  - `auteur` : l'auteur de la critique (user enregistré sur le site)
  - `texte` : le texte de la critique
  - `note` : la note (sur 100) donnée à l'album

Exercice 1 - Spécification de services. - 4 points

Nous allons ici spécifier certaines fonctionnalités des APIs de notre plateforme. Ces fonctionnalités se basent sur un paradigme REST pour l'accès aux ressources enregistrées dans les bases de données.

Exemple de spécification pour le service `Login` de connexion d'un utilisateur :
- nom : Login
- description : Permet à un utilisateur déjà inscrit de se connecter
- requête :
  - url : `/login`
  - methode : `GET`
  - paramètres :
    - `nom` : le nom de l'utilisateur
    - `password` : le mot de passe (en clair) de l'utilisateur

  - exemple : [GET] `www.metacritic.com/login?nom=arthur&password=secret`
- réponse :
  - format : JSON
  - contenu : un identifiant associé à l'utilisateur, ou une erreur
  - exemple : `{ login_id : XXX }` ou `{ error : "mot de passe incorrect" }`

Question 1

Service `NewAlbum` d'insertion d'un nouvel album.

Question 2

Service `NewCritique` d'insertion d'une nouvelle critique.

Question 3

Service `SearchCritique` de récupération de critiques. La recherche se fait soit par le nom de l'album, soit par un seuil sur le valeur de la note (soit les deux).

Exercice 2 - Bases des Données. - 4 points

Pour stocker les différentes données du site, deux bases de données sont utilisées : une base `MySQL` pour enregistrer les utilisateurs et une base `MongoDB` pour enregistrer les albums de musique et les critiques associées.

Question 1

MySQL est un système de base de données (entourez les bonnes réponses) :

(a) relationnelle

(b) non relationnelle

(c) orientée objet

(d) orientée colonne

(e) orientée document

Question 2

MongoDB est un système de base de données (entourez les bonnes réponses) :

(a) relationnelle

(b) non relationnelle

(c) orientée objet

(d) orientée colonne

(e) orientée document

Question 3

Donnez la commande SQL permettant de créer la table visant à stocker les utilisateurs.

Question 4

Ecrivez le code javascript à exécuter dans la `shell` *mongo* pour :
- insérer dans la collection `albums` l'album "Blueberry Boat" de "The Fiery Furnaces"
- insérer dans la collection `critiques` les deux critiques pour cet album :
  - "It's both an oddly comforting and exhilarating trip.", par Dot Music, note : 80
  - "A crashing disappointment." par The Guardian, note : 30

Question 5

Au lieu d'utiliser une collection séparée pour les critiques, une solution alternative aurait pu être d'enregistrer chaque critique à l'intérieur de l'album dont il se réfère. Décrivez une possible conséquence positive et une possible conséquence négative sur la manipulation et l'organisation de ces données suite à ce changement.

Question 6

Ecrivez la méthode JAVA permettant de récupérer toutes critiques qui, soit correspondent à un album donné, soit possèdent une note supérieure à un seuil donné, soit répondent à ces deux critères. Si aucun des deux critères n'est spécifié, on retourne toutes les critiques. Le retour doit contenir en plus des identifiants, les noms d'albums et d'auteurs de critique associés.

Exercice 3 - Servlets. - 4 points

Notre site repose sur Apache Tomcat, un serveur/conteneur de servlets Java.

Question 1

Par défaut, l'exécution des servlets dans Tomcat est de type «multithread». Cela signifie que chaque nouvelle requête à une servlet ___.

(a) crée un nouveau processus pour la machine virtuelle.

(b) crée une nouvelle instance de la classe servlet à l'intérieur du même thread.

(c) crée un nouveau thread sans créer une nouvelle instance de la classe servlet.

(d) crée un nouveau thread et une nouvelle instance de la classe servlet.

Question 2

Donnez le code de la servlet et des methodes associée pour le service `SearchCritique` définie dans l'Exercice 1.

Exercice 5 - Client. - 6 points

Soit le code HTML / CSS suivant : main.html :

```
<!DOCTYPE html>
<html>
  <head>
     <link href="main.css" rel="stylesheet" type="text/css">
  </head>
  <body>

            <div id="critiques">
                   <div class="critique">
                      <div class="infos_album">
                           Album = Blueberry Boat, Groupe =The Fiery Furnaces
                      </div>
                      <div class="text_critique">
                           It's both an oddly comforting and exhilarating trip.
                      </div>
                      <div class="infos">
                            <span> Posté par Dot Music </span>
                            <span> Note : 80 </span>
                      </div>
                   </div>
                   <div class="critique">
                      <div class="infos_album">
                           Album = Blueberry Boat, Groupe =The Fiery Furnaces
                      </div>
                      <div class="text_critique">
                           A crashing disappointment.
                      </div>
                      <div class="infos">
                            <span> Posté par "The Guardian" </span>
                            <span> Note : 30  </span>
                      </div>
                   </div>
          </div>
    </body>
</html>
```

main.css :

```
#critiques{
      margin-top:60px;
      padding-top:20px;
      padding-bottom:20px;
      width:80%;
      margin-left:20%;
      border:solid;
}

.critique{
      position:relative;
      margin-left:10%;
      width:80%;
      margin-top:20px;
      margin-bottom:40px;
      border:solid;
      padding: 10px 5px 10px 5px;
      height:60px;
}
.infos{
  position:absolute;
  right:20px;
  bottom:0px;
  margin-top:15px;
  font-style:italic;
}
```

Question 1

Dire ce qu'il faut faire dans main.css pour que :

- le texte de la critique observe un espacement vertical de 20 pixels avec les informations sur l'album
- le texte contenu dans la boîte contenant la note d'une critique soit en rouge

Question 2

Dire (en justifiant) ce qu'il se passe si l'on retire la ligne *position :relative* du fichier css.

Question 3

Donner ce qu'il faut ajouter à chaque boite correspondant à une critique dans le fichier html pour qu'apparaisse un lien permettant d'appeler une fonction javascript de suppression de la critique *supCritique* (penser aux arguments de cette fonction, sachant que la suppression doit être effective sur le serveur). Le code de cette fonction n'est pas demandé.

Question 4

En considérant que les communications doivent se faire en asynchrone par le biais d'AJAX et que la servlet correspondant au service SearchCritique est stockée sur le même domaine que le code client, écrire le code javascript (uniquement le code de l'appel AJAX) permettant de récupérer la liste des critiques répondant aux critères passés à la recherche (voir exercices précédents). En cas de succès de la communication, on appelera une fonction traiteReponseCritiques (à écrire dans les questions suivantes) qui attend la réponse au format texte. En cas d'echec on affichera un message "Echec de la communication avec le serveur"

Question 5

Ecrire les classes Javascript suivantes :

- Critique contenant les informations correspondant à une critique ;
- ListeCritiques contenant une liste de critiques.

Pour ces 2 classes, vous implémenterez la méthoge **getHtml()** correspondante. Vous ferez en sorte que si l'utilisateur connecté (variable `env.actif`) est proprietaire d'une critique, un lien "supprimer" soit contenu dans sa boîte HTML, et qu'un clic sur ce lien appelle la fonction javascript de suppression de la critique *supCritique* (on considère cette fonction déjà écrite).

Question 6

Écrire une fonction ListeCritiques.revival qui définit la manière de reconstruire un objet ListeCritiques à partir d'un texte au format JSON envoyé par le service SearchCritique définit précédemment. Cette fonction sera utilisée dans la fonction traiteReponseCritiques de la manière suivante :

```
var liste=JSON.parse(json_texte,ListeCritiques.revival);
```

avec json_texte correspondant au texte retourné par la servlet ListCritiques

Question 7

Écrire la fonction traiteReponseListCritiques permettant d'afficher la liste de critiques retournée par le serveur dans l'interface client

Exercice 6 - Map/Reduce - 2 points

Question 1

Donnez le code des fonctions map et reduce permettant le calcul de la note moyenne de chaque album