



Ulysse Cadour  
Alexandre Penkler

Compte Rendu Projet  
Petit Théâtre :  
Application multi-niveaux  
Mars 2012

# 1 Compléter les interactions fournies

Dans cette partie nous avons complété les deux servlets : (ProgrammeServlet et NouvelleRepresentationServlet) en garantissant bien entendu l'intégrité de la base.

## 2 Fournir de nouvelles interactions

Nous avons écrit une servlet permettant de consulter la liste des représentations disponible. Nous l'avons testé avec des requêtes concurrentes. Nous avons remarqué que le servlet interagissait correctement avec la base. Nous avons réalisé des servlet permettant de consulter l'ensemble des places disponible. L'avantage que cela nous a donné par la suite a été de pouvoir réutiliser les requêtes sur la base de données pour des requêtes d'ajout de place et de représentation dans notre panier Dynamique et Statique.

### 2.1 Intégrité

Pour garantir l'intégrité de la base nous avons utilisé deux approches. Une première qui repose sur des vérifications au sein de la base de données, par exemple que lorsque l'on réserve une place nous vérifions que la place n'est pas déjà dans les Tickets. Souvent dans ce cas nous utilisons le chaînage d'exception pour faire ressortir l'erreur.

Une autre approche était d'assurer certaines contraintes directement au niveau du logiciel, soit le passage de paramètre etc.. Par exemple c'est au niveau du logiciel que l'on vérifie qu'une représentation n'est pas dépassée avant de l'ajouter au caddie.

### 2.2 Liens entre les Servlets

Nous avons créé des liens entre les servlet de manière à utiliser les résultats d'une comme paramètres d'une autre. Ainsi cela réduit l'utilisation de formulaires ce qui est fastidieux pour l'utilisateur.

Pour ce faire nous avons utilisé deux méthodes. La première méthode est d'utiliser les cookies, ce qui est pratique car on peut stocker des structures de données entières sans avoir à les parser.

La deuxième méthode est d'utiliser la requête html. C'est à dire on appelle une autre servlet avec une requête html et on utilise "[url] ?parametre1=valeur&parametre2=valeur...". Dans ce cas on ne peut stocker que des valeurs simples et non des structures de données complexes. Par contre contrairement à la méthode précédente si que l'utilisateur bloque les cookies sur son navigateur ce n'est pas problématique.

## 3 Caddie du client

### 3.1 caddie volatile

Cette partie nous a permis de créer une structure de données qui permettra d'ajouter des représentation est des places à celles-ci. Une servlet d'ajout de places à ce caddie a été mise en place. Ce caddie stocké dans les cookies de l'utilisateur, est automatiquement détruit au bout d'un certain temps. Une servlet de validation du caddie à été mise en place afin de permettre d'ajouter les représentations a la base de données et ainsi de ne plus permettre la vente de celles-ci. Une autre Servlet à été mise en place afin de permettre de modifier le caddie. Tout en respectant les contraintes d'intégrité(Interdiction d'ajouter au caddie une place déjà réservée).

### 3.2 caddie persistant

Dans cette partie une nouvelle table panier composée de 4 champs à été ajoutée dans la base de données afin de permettre de stocker ces informations pour une utilisation future.(La date de la représentations le numéro de spectacle la place et le rang) L'utilisateur n'as pas conscience de la sauvegarde de son caddie lors de la réservation(celui-ci est transparent) Afin de réaliser cela nous avons modifié la servlet permettant d'ajouter des places au caddie. Elle contacte alors la base de donnée et lui indique les modifications de l'utilisateur sur la base.

Une servlet d'édition du panier à aussi été ajoutée afin que l'utilisateur puisse retirer des "Items" de son caddie mais cette fois-ci en mettant à jour la base de données.

Enfin, nous avons garantie la cohérence dans la base de donnée en effectuant un rollback lorsque l'utilisateur ne peut réserver la totalité des places demandées.

De plus nous avons vidé le panier lors de la validation de celui-ci cela permettant d'éviter à l'utilisateur de redemander des Places déjà réservées. Bien que la base n'accepte pas d'ajouter des places réservées et que nous avons interdit de faire de telles requêtes. Nous avons de plus mis en place un mécanisme permettant de vérifier que si deux utilisateur voulant réserver la même place ne puisse le faire. Le premier utilisateur ayant validé sont panier est alors autorisé à réserver le contenu de celui-ci.

## 4 Sécurité

Pour effectuer cette étape il nous a fallut configurer *apache-TomCat* pour protéger certaines pages. Ainsi on liste dans le fichier *web.xml* les pages disponibles uniquement aux utilisateurs logués. La restriction va plus loin en permettant de restreindre l'accès à certain type d'utilisateurs. Dans notre cas les utilisateurs logués

de type *admin*. Ensuite il faut s'assurer que dans le fichier *tomcat-users.xml* le type et un nom utilisateur est défini. Pour faciliter ce travail nous avons utilisé un dossier admin contenant les pages sensibles.

## 5 Présentation

Nous avons inclut du code java-script dans notre application afin de mettre en valeur certains éléments. Une bibliothèque java-script à été utilisée afin de visualiser le contenu de la base plus aisément. Ce code java-script est bien adapté pour des requêtes sur une base de donnée.

## 6 travail restant

Faute de temps nous n'avons pas pu remplacer les pages données en html en jsp pour pouvoir y inclure directement du code java donnant du contenu dynamique. Cela dis nous avons fait l'étude et les recherches pour savoir comment faire.