

TP n°2

La série de Tps proposée vise à simuler une partie simplifiée de « 421 ». L'opérateur jouera contre l'ordinateur. Les règles seront données au fur et à mesure des besoins.

Objectifs du TP

Séparation de l'aspect traitement (implanté par une servlet) de l'aspect présentation (implanté par une jsp).
Création de page d'erreur.

Question 1

Créer un nouveau projet `tp2`. Ce projet sera une réécriture du précédent en utilisant les JSP pour l'affichage du formulaire d'authentification et non plus la servlet.

Renommer le fichier `authentification.html` du `tp1` en `authentification.jsp`. La zone de texte `nom` est représentée dans la jsp par la variable `nom`. La valeur saisie est enregistrée dans la requête (`request`) et récupérée par la méthode `getParameter(...)`. Cette valeur est affichée à chaque chargement de la page jsp. Au premier chargement de la page, la zone de saisie contiendra la chaîne : "Votre nom". Après avoir tapé un nom, la jsp s'exécute de nouveau pour afficher ce nouveau nom.

Rappel : Balise JSP

- Les scripts sont des instructions Java embarquées dans du code HTML entre les balises `<% %>`
- Les déclarations globales à une page JSP sont effectuées entre les balises `<%! %>`
- Une balise d'expression contient une expression Java `<%= %>`
Exemple : `<input name='Nom' value= '<%= nom %>' type='text'>`

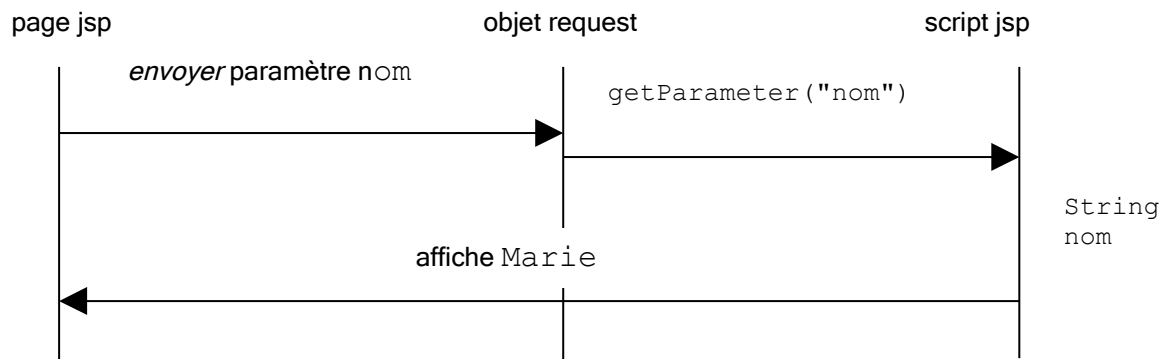
Question 2

Lancer l'application en tapant l'URL suivante :

```
http://localhost:8080/tp2/authentification.jsp
```

Que se passe-t-il ?

*Supposons que Marie soit associée au paramètre `nom` et contenu dans l'objet `request` le script jsp situé au début du fichier permet de récupérer cette valeur dans une variable java la nouvelle valeur du paramètre jsp `nom` prend la valeur Marie.
L'objet `request` est un objet implicite. Il n'est donc pas déclaré.*



Question 3

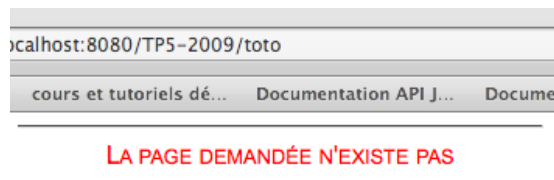
On souhaite maintenant donner une meilleure présentation plus adaptée.

On définit le code d'erreur 404 qui correspond à une page non trouvée. Le fichier `web.xml` est complété de la manière suivante où `/404.html` est le fichier qui traite l'erreur en affichant le message : **"La page demandée n'existe pas!"**.

```

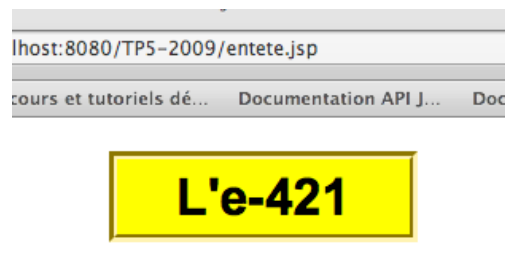
<error-page>
  <error-code>
    404
  </error-code>
  <location>
    /404.html
  </location>
</error-page>
  
```

Lancer à nouveau l'application en tapant l'URL suivante : <http://localhost:8080/tp2/toto>



Question 4

Créer le fichier `entete.jsp` capable d'afficher une bannière. Le style de la bannière est laissé à votre initiative.



Pour que les futures pages de l'application partagent toutes la même bannière, il suffit de modifier la version précédente du fichier `authentification.jsp` et d'y inclure le fichier `entete.jsp`.

Question 5

Création du package `entites`

Chaque joueur est caractérisé par ses nom, prénom et age. Initialement, chaque joueur possède 12 jetons. Chaque joueur est une instance de la classe `entites.Joueur`. L'ensemble des joueurs est mémorisé dans la classe `Joueurs` sous la forme d'une collection de type `Hashtable`. On considérera que la clé est le nom du joueur (pas de synonymie sur les noms)

Créer les classes `Joueur`, `Joueurs` du package `entites`.

Question 6

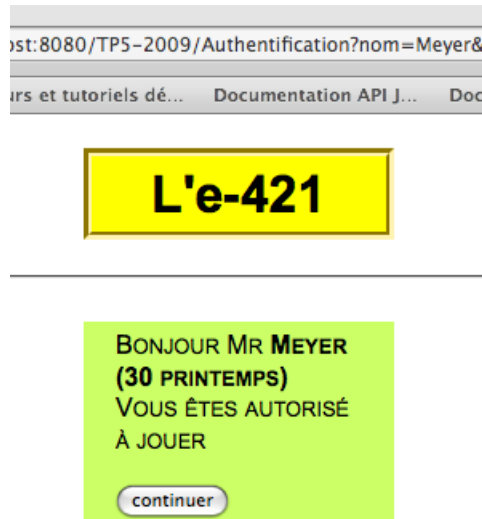
Modification de la servlet `Authentification`

La servlet `Authentification` crée et enregistre les joueurs dans une instance de la classe `Joueurs`:

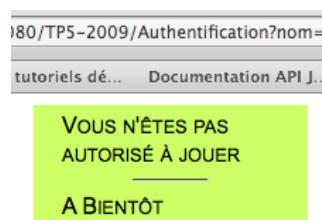
Meyer,Luc,30
Dupont,René,40
Legrand,Lisa,16

La servlet `Authentification` n'affiche plus de réponse comme dans le TP1. Elle laisse ce rôle à la JSP `accueil.jsp`. Après réception du nom à authentifier, la servlet vérifie que ce joueur est bien autorisé à jouer (un joueur est autorisé à jouer s'il a plus de 16 ans). Dans ce cas elle transmet le contrôle à la `accueil.jsp`.

Construire cette nouvelle jsp.



S'il n'est pas autorisé à jouer, le contrôle est passé à la JSP `auRevoir.jsp`, qui affiche la fenêtre suivante :



Rappel 1 : transmission du contrôle

```
// JSPAccueil est l'URL Pattern pour accueil.jsp défini dans web.xml
RequestDispatcher
    dispatcher=request.getRequestDispatcher("JSPAccueil");
    dispatcher.forward(request,response);
```

Rappel 2 : l'objet `session` permet de mémoriser des attributs et de les rendre disponibles pour toutes les servlets/jsp pendant la durée de la session.

Dans les servlets, il est obtenu par :

```
HttpSession session = request.getSession();
```

Dans les JSP, c'est un objet implicite.