POLYTECH ANGERS

TP 4A - Java Entreprise Edition.
Création de servlets (get et session)

Nicolas Delanoue et Sylvain Joyeau

Le but de ce TP est que vous vous familiarisez avec la notion de servlet et leur déploiement dans un serveur d'application *Tomcat*.

Exercice 1 (Première servlet)

1. A l'aide d'Eclipse, créez un projet Dynamic Web Projet `MonProjet`.
2. Ajoutez à ce projet une servlet nommée `PremiereServlet`.
3. Complétez la méthode `doGet()` de sorte qu'elle génère la sortie html suivante :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
     <p> C'est ma première servlet !</p>
  </body>
</html>
```

4. Testez l'exécution de votre servlet depuis Eclipse.
5. Modifiez l'annotation adéquate de sorte que votre servlet s'exécute lors d'une demande d'accès à la ressource nommée `bonjour`.
6. Déployez et testez cette servlet depuis votre navigateur Firefox.
7. Dans Firefox, un appui sur la touche F12 nous permet d'en savoir plus sur les échanges entre le client et le serveur. Videz le cache récent de Firefox avec la combinaison de touche Ctrl + Shift + Suppr, et rechargez la page. Depuis l'onglet Réseau, combien de requêtes http get votre navigateur a-t-il émises ?

Exercice 2 (Récupérer des données transmises par le client)

L'indice de masse corporelle (IMC) permet d'évaluer rapidement votre corpulence simplement avec votre poids et votre taille, quel que soit votre sexe. Il se calcule simplement en divisant le poids (en kilogramme) par le carré de la taille (en mètre).

1. Créez une servlet nommée `CalculDeMonImc` qui affichera l'indice de masse corporelle sur l'écran du navigateur lorsque l'url suivante est invoquée
   `http://VotreIP:LeBonPort/CalculDeMonImc?poids=94&taille=1.86`
2. Créez un formulaire html statique nommé `renseignement.html` qui lors de sa soumission fait appel à la servlet précédente en get. Ce formulaire pourra avoir l'apparence de celle donnée sur la figure 1.

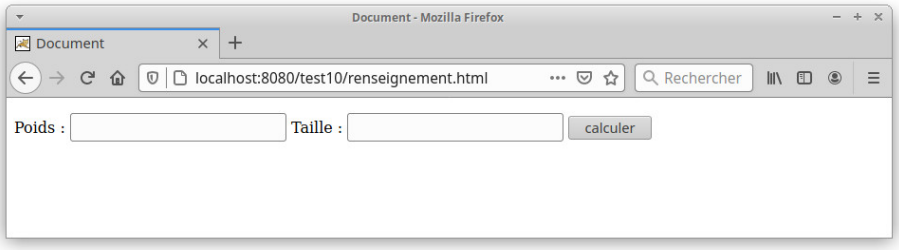

FIGURE 1 – Formulaire de saisie

3. Testez dans Firefox et surveillez les échanges entre le client et le serveur lors d'une soumission.
4. Reprenez à partir de la question 2 de sorte que la soumission s'appuie sur une commande http post. Comment devez-vous modifier votre servlet pour que le calcul s'opère correctement ?
5. Quelle différence existe-il encore http get et http post ?

Exercice 3 (Une classe "métier" dans notre application web)
Dans cet exercice, nous allons commencer à découpler la partie business (logique) de la partie vue (affichage).

1. Créez une classe java nommée `Imc`.
2. Ajoutez les deux attributs privés suivants à cette classe : `taille` et `poids` qui seront de type `double`.
3. Ajoutez un constructeur avec comme prototype :

```
public Imc(double taille, double poids);
```

4. Ajoutez une méthode qui calcule l'imc avec le prototype suivant :

```
double calcul();
```

5. Modifiez la servlet `CalculDeMonImc` pour qu'elle s'appuie sur un objet de la classe `Imc` afin de faire le calcul.
   (a) Où avez-vous déclaré votre objet ?
   (b) dans quelle méthode l'avez construit ?
   (c) quelle est la portée de cet objet ?

Exercice 4 (Cookies et Sessions)
Cet exercice est dans la continuité du précédent.

1. L'objectif de cette question est de créer une page `/TableauDeBord` qui affiche les données (poids, taille, imc), pour ce faire, vous devez :
   — ajouter, dans la servlet `CalculDeMonImc`, une écriture de cookies,
   — ajouter, dans la servlet `/TableauDeBord`, une lecture des cookies..
2. Avec Firefox, scrutez les échanges entre votre navigateur et le serveur. Quand les cookies sont-ils transmis ? (Indication : allez voir dans Stockage).
3. Que se passe-t-il si vous allez directement chercher la ressource `/TableauDeBord` avec un autre navigateur (Microsoft Edge ou bien en navigation privée si vous n'avez pas d'autre navigateur).

4. Où sont stockées ces données ? Que se passerait-il en cas d'arrêt du serveur d'applications ?
5. Reprenez les questions précédentes avec l'aide de la classe `HttpSession` en créant des servlets nommées `/TableauDeBord2` et `/CalculDeMonImc2`. Est-il possible de sauvegarder un objet de la classe `Imc` dans une session ?
6. Du point de vue du développeur web, laquelle de ces technologies est la meilleure pour sauvegarder ces données ? même question d'un point de vue réseau et sécurité ?

Exercice 5

L'objectif de cet exercice est de créer une page web qui permet à un groupe d'amis de construire une liste commune de bandes dessinées.

Techniquement, nous allons développer une servlet qui traite un formulaire et affiche cette liste.

1. Créez une servlet http nommée `/BandesDessinees`.
2. Ajoutez à cette classe la propriété suivante :

   ```
   private ArrayList<String> listeBandesDessinees;
   ```

3. Dans quelle méthode héritée est-il pertinent de placer l'instruction suivante pour respecter l'objectif de l'exercice :

   ```
   listeBandesDessinees = new ArrayList<String>();
   ```

4. Lorsque la servlet répond à la commande http GET, faite en sorte qu'elle affiche le formulaire de saisie d'une bande dessinée, avec deux boutons, un pour Ajouter, l'autre pour Retirer (cf figure 2).

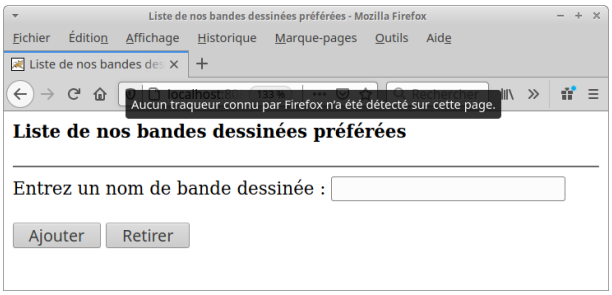

FIGURE 2 – Formulaire de saisie

Le code html suivant doit être celui généré.

```
<html>
  <head><title>Liste de nos bandes dessinées préférées</title></head>
  <body>
    <h4>Liste de nos bandes dessinées préférées </h4>
    <form method="POST" action="BandesDessinees">
    Entrez un nom de bande dessinée :
    <input type="text" name="bd"><br/><br/>
    <input type="submit" name="action" value="Ajouter">
    <input type="submit" name="action" value="Retirer">
    </form>
```

```
  </body>
</html>
```

5. Modifiez votre servlet de sorte qu'elle réponde à une commande http post. Elle devra itérativement
   (a) récupèrer l'information concernant le bouton qui a été sélectionné (Ajout ou Retrait),
   (b) ajouter ou supprimer la bande dessinée dans la liste,
   (c) puis afficher si l'ajout ou la suppression se sont correctement déroulés (cf Figure 3).

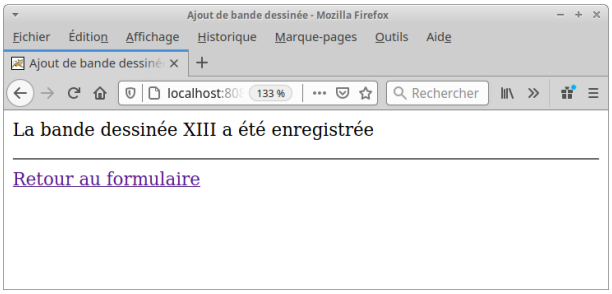

FIGURE 3 – Interface de confirmation.

6. Quelle méthode de la classe `HttpServletRequest` permet de connaitre le nom de la ressource actuelle, utilisez cette méthode pour le lien associé au "Retour au formulaire" ?
7. Modifiez la méthode `doGet()` de sorte que la liste des bandes dessinées s'affiche au dessus du formulaire. Voir figure 4.

FIGURE 4 – Affichage de la liste au dessus du formulaire de saisie.

8. Gérez les doublons afin qu'une bande dessinée ne puisse être présente qu'une fois dans la liste.
9. Est ce que votre application construit une liste par utilisateur ou bien une liste commune à tous les utilisateurs ?
10. Lors de la fermeture de votre navigateur, est-ce que la liste est sauvegardée ?
11. Lors de l'arrêt de la servlet (ou du serveur d'applications), pourquoi est-ce que la liste disparaît ?
12. Quel mécanisme pourrait être mise en oeuvre pour sauvegarder cette liste même lors d'un redéploiement ?

Exercice 6 (Pour aller plus loin : les filtres)
L'objectif de cet exercice est de mettre en place une filtre sur des servlets. L'idée principale des filtres est de lancer du code en amont et en aval de l'exécution d'une servlet. La figure 5 permet d'illustrer un exemple de suite d'appels lors de la mise en place d'un filtre sur une servlet.

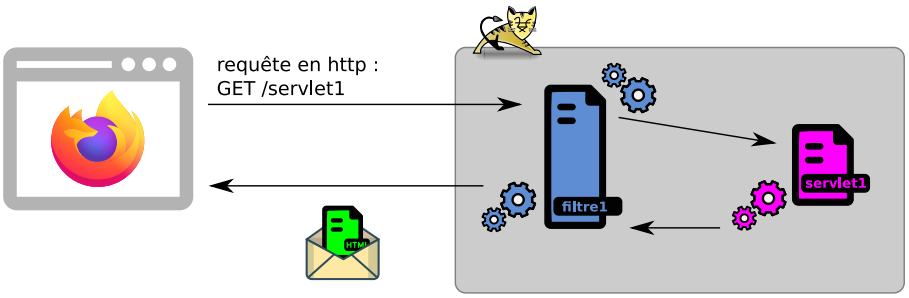

FIGURE 5 – Illustration de la place d'un filtre dans la chaîne de traitements.

D'un point de vue programmation, la mise en place et le cycle de vie d'un filtre ressemble à une servlet. Ces filtres sont classiquement utilisées dans les contextes suivants :

- Journalisation des paramètres de demande dans les fichiers de log.
- Authentification avant la demande de ressources.
- Formatage du corps ou de l'en-tête de la requête avant de l'envoyer à une servlet.
- Compression des données de réponse envoyées au client.
- Modification de la réponse avec l'ajout de cookies, d'informations d'en-tête, ...

1. Renseignez-vous sur la classe `Filter`, et ses méthodes `init()`, `doFilter()` et `destroy()`.
2. Renseignez-vous sur l'annotation `@WebFilter`.
3. Dans le contexte de l'exercice 4, mettez en place un filtre de sorte que l'accès à `/TableauDeBord` ne puisse se faire uniquement si l'utilisateur a déjà une variable de session contenant les données poids et tailles.
   Indications : on pourra utiliser la méthode `sendRedirect()` de la classe `ServletResponse` pour rediriger un utilisateur vers la bonne ressource en cas d'échec.
4. Peut-on mettre un filtre pour un ensemble d'URIs ?