

ING2
Semestre 1

Développement d'applications Web

Éric Pipard

Travail dirigé n° 3
Développement d'une Servlet

Objectifs : maîtriser la création de Servlets

Mots-clés : Servlets, URL, HTTP, session, contexte des webapp

Table des matières

1 – Une servlet dans un formulaire avec GET.....	3
2 – Une servlet dans un formulaire avec POST.....	4
3 – Une simple servlet.....	5
4 – Le mécanisme des sessions.....	6
5 – Exploitation du contexte d'une webapp.....	6

1 – Une servlet dans un formulaire avec GET

Les servlets permettent de gérer des requêtes HTTP et de fournir au client une réponse HTTP.

Les servlets s'exécutent toujours dans un moteur de Servlet ou conteneur de Servlet permettant d'établir le lien entre la Servlet et le serveur Web.

Une Servlet est invoquée lorsqu'un navigateur appelle l'URL qui lui est liée.

Nous allons créer une page HTML qui sera exécutée lorsque le navigateur demandera à charger la page formulaire.html et qui exécutera une servlet nommée GetInfo

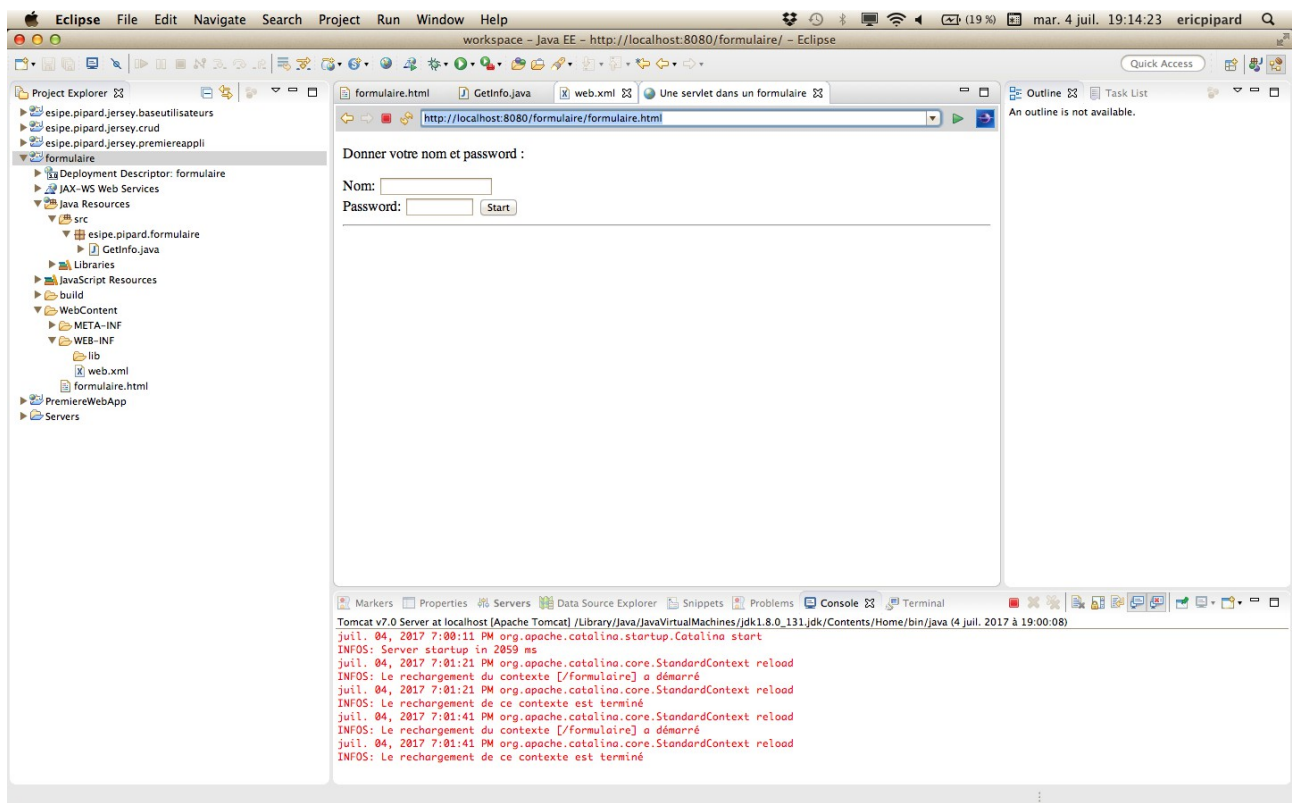
La servlet dérive de la classe `HttpServlet`, appartenant au package `javax.servlet.http.HttpServlet` ; la servlet gère donc la réception de requêtes HTTP et l'envoi de réponses HTTP.

Lorsque le navigateur client demande le chargement d'une page, le serveur Web Tomcat reçoit une requête HTTP de type GET ou POST. Il crée un objet de type `HttpServletRequest`, qui pourra ensuite être traité par la Servlet. Il en va de même pour la réponse : c'est Tomcat qui se charge de transformer l'objet `HttpServletResponse` manipulé par la Servlet en une réponse HTTP compréhensible par le navigateur.

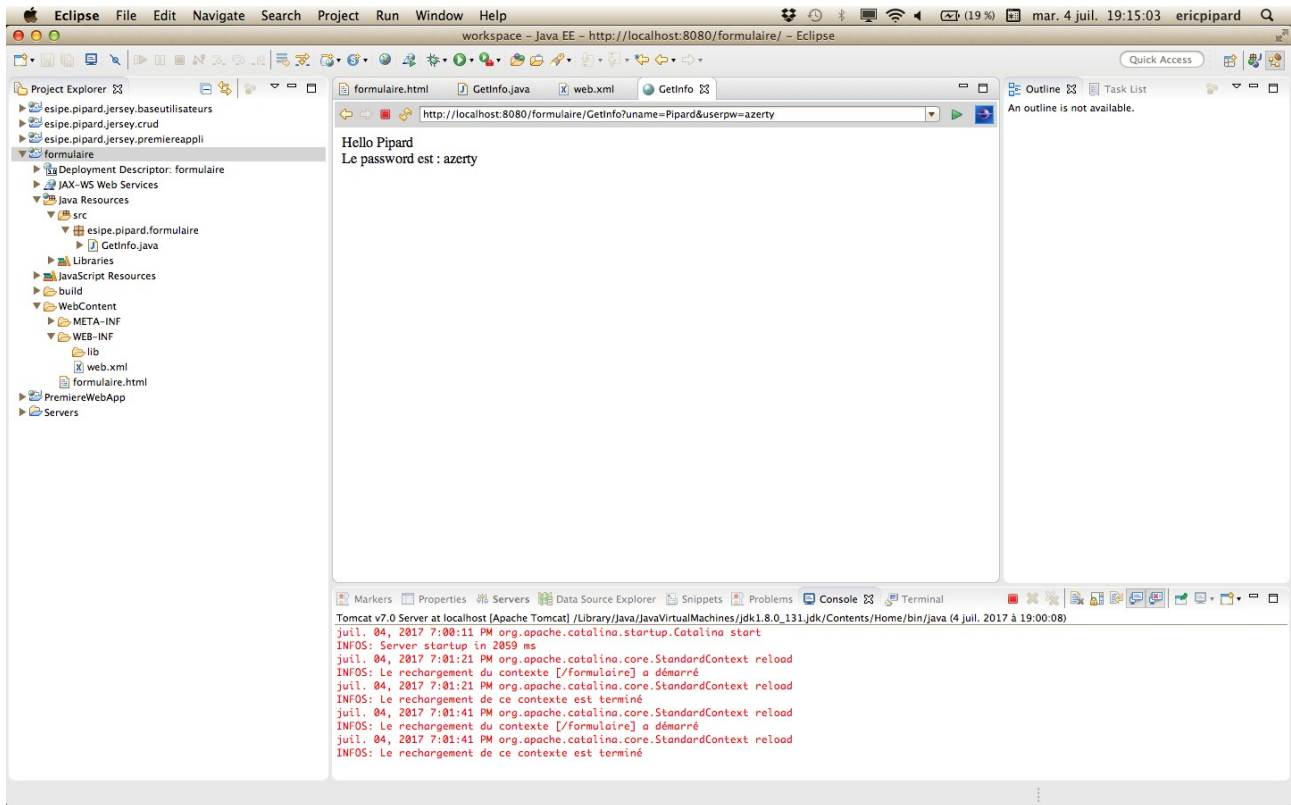
Si la servlet a été appelée par la méthode GET, la méthode `doGet` est appelée pour traiter la requête du client. Si la servlet a été appelée par la méthode POST, la méthode `doPost` est appelée pour traiter la requête du client.

L'objet `HttpServletRequest` fournit une méthode pour récupérer les valeurs dans un formulaire : `getParameter("uname")` avec `<input name="uname" ... >`.

Voici le formulaire :

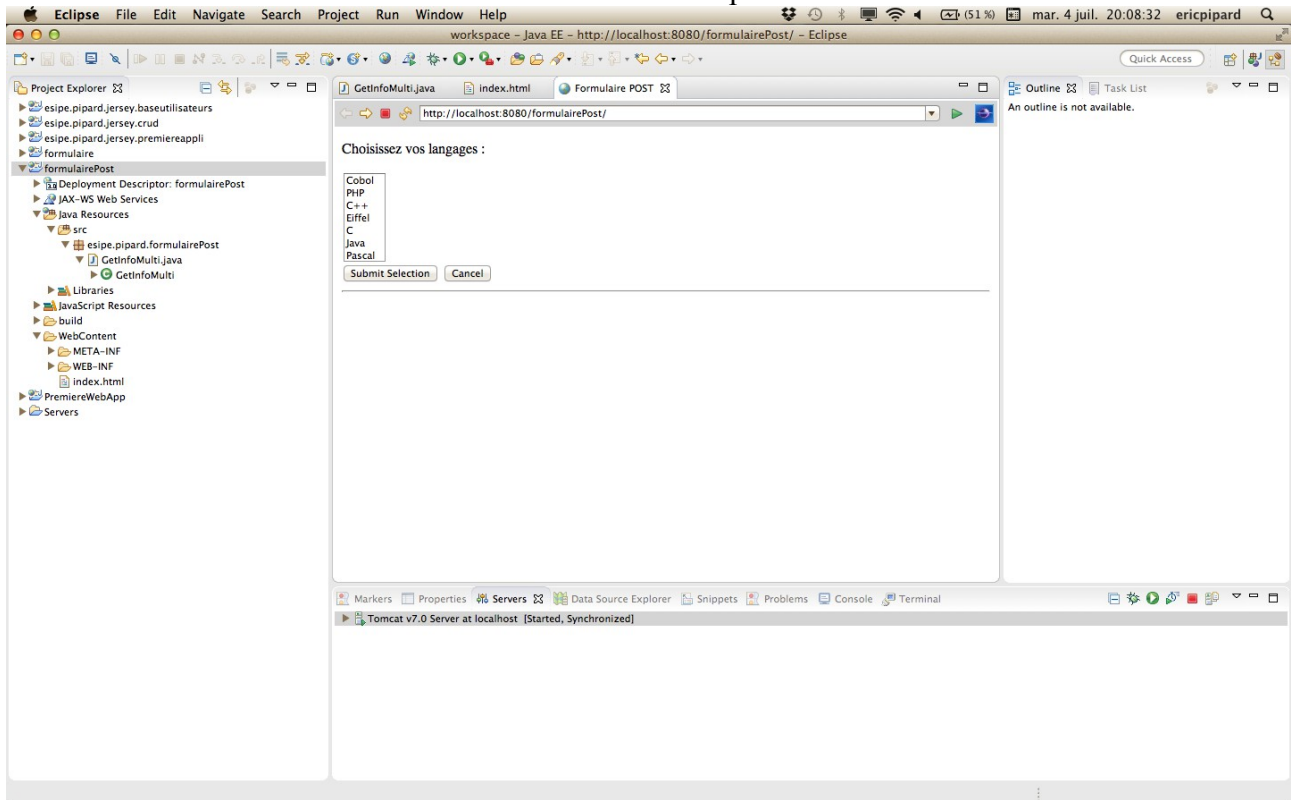


Et le résultat :

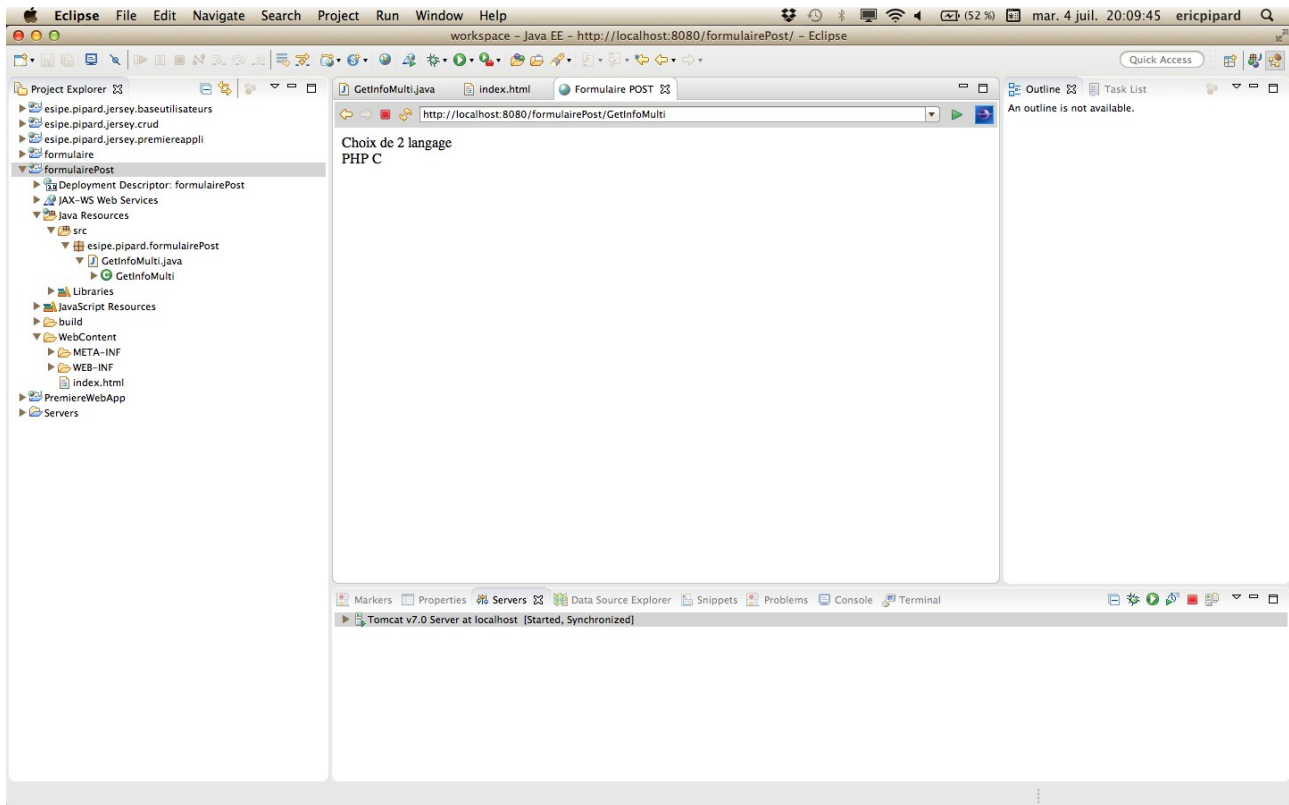


2 – Une servlet dans un formulaire avec POST

Vous devez créer un formulaire avec une sélection multiple. Voici le formulaire :



Et le résultat :

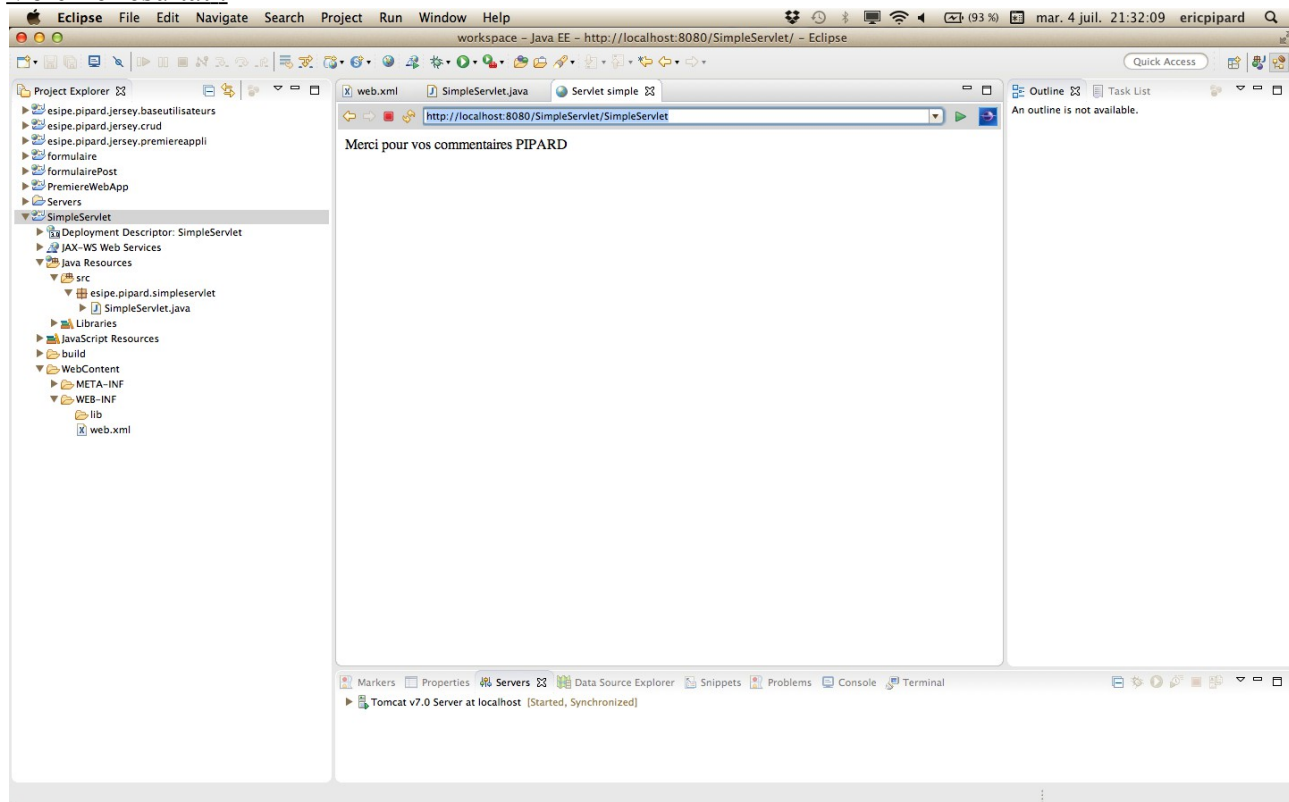


L'objet `HttpServletRequest` fournit une méthode pour récupérer un tableau de string dans un formulaire : `getParameterValues("topic")` avec `<select name="topic" multiple>`.

3 – Une simple servlet

Créer une servlet permettant d'écrire dans un fichier. Ce dernier sera indiqué dans le fichier `web.xml` avec `<init-param>`.

Voici le résultat :



Le fichier :

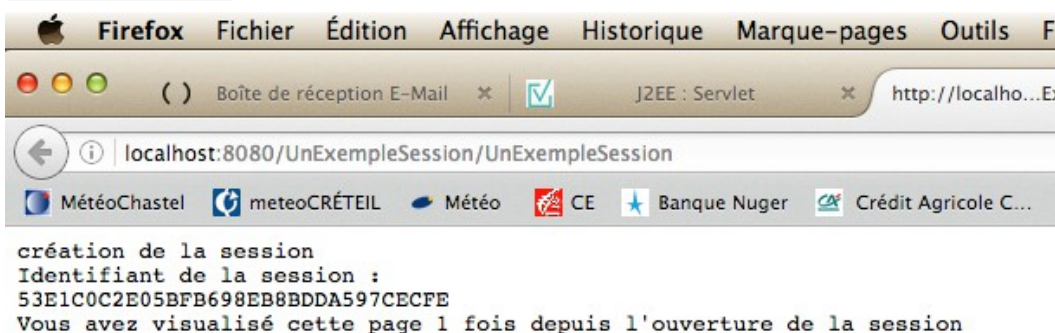


```
User: PIPARD dit Je suis le meilleur
User: PIPARD dit Je suis le meilleur
User: PIPARD dit Je suis le meilleur
User: PIPARD dit Je suis le meilleur
User: PIPARD dit Je suis le meilleur
User: PIPARD dit Je suis le meilleur
User: PIPARD dit Je suis le meilleur
```

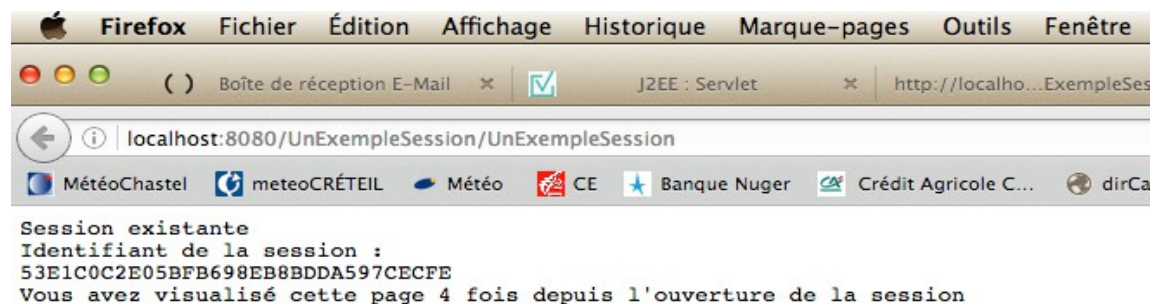
4 – Le mécanisme des sessions

Le principe des sessions permet le partage d'informations entre plusieurs pages JSP ou servlet. Créer une servlet permettant d'indiquer au client le nombre de fois qu'il a appelé la dite servlet durant la session ; il faudra utiliser le paramètre `HttpServletRequest` de la méthode `doPost` afin de récupérer la référence de la session. Actualiser la page à plusieurs reprises sous un navigateur ; le compteur doit s'incrémenter.

Résultat attendu :



création de la session
Identifiant de la session :
53E1C0C2E05BFB698EB8BDDA597CECFE
Vous avez visualisé cette page 1 fois depuis l'ouverture de la session



Session existante
Identifiant de la session :
53E1C0C2E05BFB698EB8BDDA597CECFE
Vous avez visualisé cette page 4 fois depuis l'ouverture de la session

5 – Exploitation du contexte d'une webapp

Le contexte d'une application web permet de partager des données. Les données restent disponibles en permanence (au delà de la durée de visite du site web ie fermeture du navigateur) contrairement aux sessions (recharger la servlet sous Eclipse).



Créer une servlet permettant de présenter aux clients les informations suivantes :

- le nombre de fois que la servlet a été utilisée depuis le lancement de l'application web
- une horodate de dernier appel à la servlet

Le contexte d'une webapp, attribut de toute servlet, est une implémentation de l'interface `javax.servlet.ServletContext`.

Résultat attendu :

