Maintenir un état entre deux requêtes

Philippe Mathieu & Guillaume Dufrene

IUT-A Lille
http://www.iut-a.univ-lille.fr
prenom.nom@univ-lille.fr

Cycle de Vie

Université de Lille

- Une Servlet est un objet Java
- Une seule instance est créée par le conteneur à la première invocation

- 1 appel au constructeur de la servlet
- 1 appel à la méthode `init` de la servlet
- n appels à la méthode de service (`service`,`doGet`,`doPost`,`doPut`,`doDelete`)
- 1 appel à la méthode `destroy` en fin de vie

La fin de vie est déclenchée par le serveur, soit en cas d'arrêt, soit en cas d'inutilisation trop longue

Exemple

Université de Lille

```java
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;

@WebServlet("/Vie")
public class Vie extends HttpServlet
{
  public void init(ServletConfig config)
  throws ServletException
  {
    super.init(config);
    System.out.println("Démarrage de la servlet");
  }

  public void service(HttpServletRequest req, HttpServletResponse res)
  throws ServletException, IOException
  {
    res.setContentType("text/html;charset=UTF-8");
    PrintWriter out = res.getWriter();
     out.println("Exécution de la méthode de service");
  }

  public void destroy()
  {
    super.destroy();
    System.out.println("Arrêt de la servlet");
  }
}
```

Le conteneur JEE

Université de Lille

Pour être exécutée, une servlet doit être placée dans un conteneur JEE

- Gère les contextes
  les contextes sont indépendants et peuvent être démarrés ou arrêtés séparément
- Gère les servlets
  - l'invocation des méthodes demandées par les clients
  - le multi-threading
    Chaque requête s'exécute dans un thread différent
  - durée de vie des servlets
    les servlets sont persistantes
  - Sécurité

HTTP un protocole sans état

Université de Lille

- HTTP est un mode sans état (Contrairement à FTP ou Telnet)
- Après avoir "servi" une page, la connexion est rompue
- Le serveur web ne maintient pas les informations à propos du client entre deux requêtes
- le serveur ne sait pas déterminer si une requête ou une réponse provient du même client.
- De nombreuses applications nécessitent pourtant d'identifier les requêtes venant du même utilisateur pour conserver un état entre ces requêtes.

Un problème concret

Université de Lille

On souhaite avoir une page Web qui affiche

Vous avez accédé X fois à cette page sur les Y accès au total.

Une première ébauche

Université de Lille

```java
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;

@WebServlet("/Cpt")
public class Cpt extends HttpServlet
{ private int cpt=0;

  public void service(HttpServletRequest req, HttpServletResponse res)
  throws ServletException, IOException
  {
    res.setContentType("text/html;charset=UTF-8");
    PrintWriter out = res.getWriter();
    cpt++;
    out.println("Valeur du compteur : " + cpt );
  }
}
```

Comment gérer le compteur individuel ?

Université de Lille

- Champs cachés
  `<input type=hidden name="cle" value="mavaleur">`
- Concaténation d'URL
  Ajouter l'identifiant du client à la fin de chaque lien
  `<a href="mypage?cle=mavaleur">...</a>`
- Les cookies
  La première page crée le cookie chez le client
  Les pages suivantes accèdent à ce cookie préalablement.

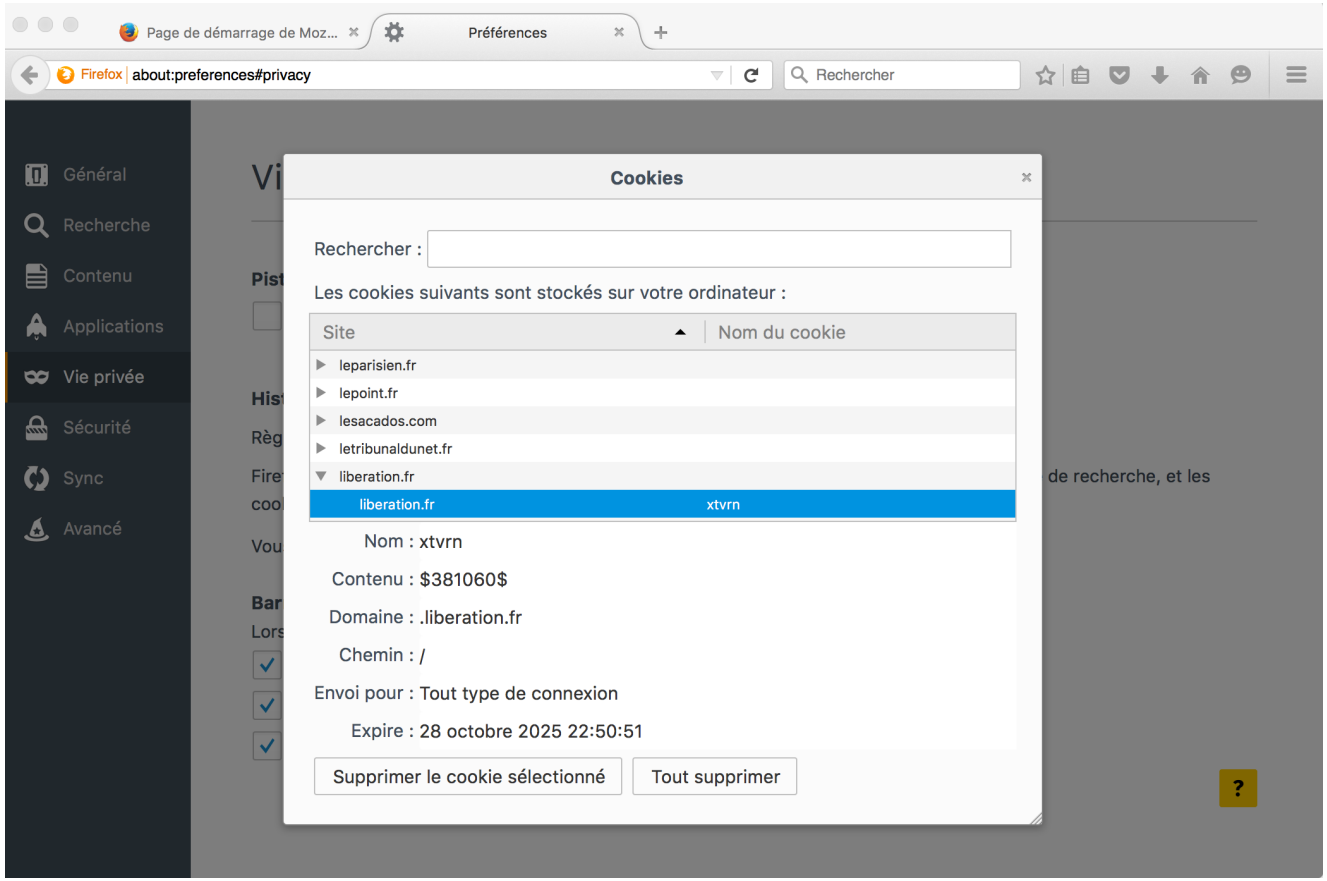
Inconvénient pour tous : Fonctionne uniquement pour des données textuelles (pas d'objet). Pb de confidentialité !.

Comment voir mes Cookies ?

Université de Lille

Firefox `Preferences -> Vie Privee -> supprimer les cookies`

Safari `Preferences->Confidentialité->Details`



Comment manipuler les cookies ?

Université de Lille

L'en-tête HTTP réservée à l'utilisation des cookies s'appelle Set-Cookie, simple ligne de texte de la forme :

```
Set-Cookie : NOM=VALEUR; domain=NOM_DE_DOMAINE; expires=DATE
```

```
// Creer
Cookie MonCookie = new Cookie("nom", "valeur");
response.addCookie(MonCookie);

// Lire
Cookie[] cookies = request.getCookies();
for(i=0; i < cookies.length; i++)
{
    Cookie MonCookie = cookies[i];
    if (MonCookie.getName().equals("moncookie"))
    {
        String Valeur = cookies[i].getValue();
    }
}
```

Inconvénients

Université de Lille

- Toute l'information est coté client
- Toutes ces solutions augmentent la longueur de la requête HTTP
- Peut nuire à la sécurité du site

En résumé

Université de Lille

- Une servlet est un objet persistant dans le serveur
- Chaque étape de sa durée de vie est accessible via une méthode
- HTTP est un protocole sans état. Il n'offre aucun moyen de passer de l'information entre deux requêtes du même utilisateur
- Les cookies offrent un moyen de stocker de l'information coté client.

La notion de session nous permettra de stocker de l'information individuelle côté serveur